gamesauce

ISSUE 2 • SPRING 2010 | GAMESAUCE.ORG

John Romero Wants to Make You A Star

Casual Games Head East
Eastern Europe Becomes Prime Location for Development and Distribution

Tales from the Trenches
Three Developers Share Beers and War Stories

Blast from the Past
Ensemble Figures Out How to Go From Empires to Kings

India's Leading Game Development Studio

It is not hard to find the best games development partner ..

You just have to

<<www.chayowogames.com

CHA YO WO ™ games

Android games | iPhone games | Virtual worlds | Facebook games | Downloadable games | Online games

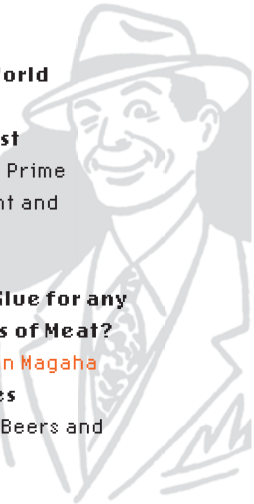
gamesauce

ISSUE 2 | SPRING 2010

COVER STORY

COLUMNS

FEATURES



Cover photograph by Thomas Hall
www.thomashallphotography.com

MISSION STATEMENT

Gamesauce is for those who have already discovered the great secret about making games: it's fun, it's cool and you get paid to do it. In publishing this magazine, we make the following promises:

- **Gamesauce** will be fun and cool—just like making games.
- We will give you a 30,000-foot view of the gaming industry so that you can see as clearly where it's going as you can where it's been.
- We will give you the good, the bad, and the ugly of the game industry, leaving you free to form and express your own opinions.
- We will not merely be interesting; we will be provocative. We will not shy away from asking the awkward question or printing the controversial answer.
- We will not spend our pages giving away source code for particle systems, but we will give you the history of them. Or we would if that wasn't quite so boring.
- We will never take ourselves too seriously. Seriously.

Please, keep your bathroom visits down to one article at a time. Other people have gotta go you know?

- Jake Simpson, editor in chief, editor@gamesauce.org

Tile art by *Invader*, photograph by *Ethan Clark*

LETTER FROM THE PUBLISHER

If you've spent any time in the Pacific Northwest whatsoever, you totally get the phrase "can't see the forest for the trees." In a place like that, in fact, sometimes you almost can't see the *trees* for the trees—or so it seems.

I bring this up because when you're as busy as we all are, with so much going on around us, it's easy to lose sight of the bigger picture. You get so preoccupied with the screwed up economy, squabbles between publishers and developers, indie opportunities and freelance opportunities and start-up opportunities and the half dozen cool ideas that you never have time to get to that you forget—*forget*—that you make games for a living—and that is freaking *awesome*.

Because, of course, you could be programming banking software instead. Or creating line drawings for instructional manuals. (Unless, of course, you were making line drawings for *Fight Club*-style airline emergency cards which would also be freaking *awesome*—but I digress.) But you're not. You're making *games*, which makes you one of the lucky ones.

If you were making movies, that would be pretty cool too, but movies are passive cool. They're effortless, arm's-length entertainment. Games, by contrast, are active, immersive, *addictive*. We create the settings, define the rules that govern whole universes, and then turn people loose in them. Our imaginary worlds become part of some people's realities. That's a powerful position to be in. It takes *cool* to a whole new level.

Which brings me, finally, to *Gamesauce*.

This issue of *Gamesauce* is intended to get you to stop *doing* for a few minutes so that you might spend a little time *thinking*. Thinking about your place and your purpose and your passion. Thinking about your various opportunities to make a lasting impact on others. Thinking about some bigger questions than *Where are we going for lunch today?*

Questions like these:

- How can I make my job, my team, my studio better? How can I make all my other developer friends wish they worked here? I recently ran into an old boss who thanked me for "being nice when I needed it most." What will your co-workers say about you ten years from now?
- What can I do to create positive change in the industry? Is there some way that I can make a difference for all of us? Am I going to continue to whine about the crap I don't like or am I willing to do something to make things better?
- Is there a way I can use this freaking awesome job to influence the broader social conversation? Or for that matter, what can I put into my next game that will somehow change the world-view of the people who play it? While I'm not suggesting we all start making games about peace, love, and flowers (please), I do think it's possible to make someone think critically about an important issue or at least to be nicer to their mothers. That would be cool too, wouldn't it?

So there you have it. We deliver to you a magazine that we think is entertaining and all the while our not-so-secret purpose is to make you want to change the world. Or something like that. That may be a bigger forest than you ever thought was out there, but we figured if we were going to kill a few trees printing this thing we had better give you a clearer perspective in doing so.

So read. Enjoy. And then after you've pondered a little, let us know what you think.

Jessica

Jessica Tams, *publisher*

gamesauce

PUBLISHER
Jessica Tams
EDITOR IN CHIEF
Jake Simpson

EDITOR
Peter Watkins
COPY EDITORS
Tennille Forsberg, Catherine Quinton

CREATIVE DIRECTOR
Gaurav Mathur
ART DIRECTOR/DESIGNER
Shirin Ardakani

COMIC ARTIST
Shaun Bryant
COMIC WRITER
Ed Kuehnel

CONTRIBUTORS
Brenda Braithwaite, Tatiana Chernova, Ian Fischer, Kenn Hoekstra, Ilari Lehtinen, Dan Magaha, Rhianna Pratchett, Bill Slease

CONTACT US
Address Changes or Removal:
Tennille Forsberg, tennille@gamesauce.org
Article Submissions:
Jake Simpson, editor@gamesauce.org
Publisher:
Jessica Tams, jessica@gamesauce.org

TRADEMARKS
© 2010 Mastermind Studios LLC.
All rights reserved. Reproduction in whole or part of this magazine is strictly prohibited. Gamesauce and the Gamesauce logo are trademarks or registered trademarks of Mastermind Studios LLC. All other product and company names mentioned herein may be trademarks of their respective owners.

DISCLOSURES
Mastermind Studios LLC's ("Mastermind") Gamesauce ("Magazine") is for informational purposes only. The Magazine contains current opinions which may change at any time; furthermore, Mastermind does not warrant or guarantee statements made by authors of articles in the Magazine. While the information included in the Magazine is derived from reliable sources, the reader is responsible for verification of information enclosed in this Magazine and Mastermind does not guarantee or warrant the accuracy or completeness of the information.

USAGE
Companies inside of the entertainment business may use information in this Magazine for internal purposes and with partners and potential partners. Members of the press may quote the Magazine.

SUBSCRIPTIONS
Gamesauce is available to professional game developers and members of the press. Professional game developers may visit *http://www.gamesauce.org/* to sign up for a free subscription. Gamesauce is postal mailed to over 19,000 game industry professionals and 2,500 members of the press.

CONTRIBUTORS

Brenda Brathwaite

Brenda is a veteran game designer and artist who's been making games since 1981 and most recently won the Vanguard award at IndieCade for her non-digital game Train. She has worked on classic series such as Wizardry and Jagged Alliance as well as the Dungeons & Dragons franchise. Presently the Creative Director at Slide, she works on social media games and entertainment products. Brenda also likes shoes and fast cars probably a little too much. Brenda can be reached at brenda.brathwaite@gamesauce.org.

Ilari Lehtinen

Ilari is a visual effects artist at Bugbear Entertainment in Finland, with an insatiable urge to find out what makes really good games really good in those unexplainable good ways. After 9 years in the industry, he's still looking for answers, which may or may not be encouraging to some. This article first appeared in slightly tackier form in the-cow.net, blogging all sorts of game and RPG shenanigans. Ilari can be reached at ilari.lehtinen@gamesauce.org.

Dan Magaha

Dan is the Studio Director for Seamless Entertainment in Austin, TX, where he lives with his wife and two children. He has been making fun since 1999 for companies such as Firaxis Games, NCsoft, and 2K Games. He blogs infrequently on production topics at thehyperbolist.org and spends too much time bemoaning the sad state of Washington, D.C. area sports on forums.

Rhianna Pratchett

Rhianna is an award winning writer and story creator. The games she's worked on include: Heavenly Sword, Mirror's Edge, Viking: Battle for Asgard, Risen and the entire Overlord series. She lives in London with a couple of neurotic tabby cats and a gay tortoise. True to British form, she drinks a lot of tea. Rhianna's website is rhiannapratchett.com and she can be reached at rhianna.pratchett@gamesauce.org.

Ian Fischer

Ian joined Ensemble Studios in 1997 as an Assistant Designer on Age of Empires. He was involved with every game released by Ensemble. Lead Designer credits include Age of Kings and Age of Mythology. Following the closure of Ensemble in 2009, he helped found Robot Entertainment. He has a degree from the University of Pittsburgh, served as a rescue swimmer in the United States Navy, and his garage is filled with parts that will one day be a car again. Ian can be reached at ian.fischer@gamesauce.org.

Kenn Hoekstra

Kenn has contributed to nearly twenty PC and console titles since 1997 as a Producer, Project Administrator and Level Designer for Raven Software in Madison, WI and Pi Studios, LP in Houston, TX.

Kenn has also penned a strategy guide, several game dev articles, multiple product manuals and the screenplay for Soldier of Fortune II: Double Helix. You can find his rantings and pontifications online on Twitter and at kennhoekstra.com.

Tatiana Chernova

Tatiana joined Alawar Entertainment in 2004. Initially, she was responsible for game localizations and translations. Today, Ms. Chernova is director of marketing communications and PR. Her responsibilities include soliciting media coverage of the company's products and the creation and launch of PR campaigns. Ms. Chernova is also an advisor to the Casual Games Association. She holds a master's degree in linguistics and a second master's degree in economics and management from Novosibirsk State University. Before coming to Alawar, Ms. Chernova worked as a TV journalist and hosted a program featuring notable stories about city life in Russia. Tatiana can be reached at tatiana.chernova@gamesauce.org.

Bill Slease

Bill came to the video game industry after working as a Senior Research Programmer at Carnegie Mellon's Center for Cognitive Brain Imaging where he built control systems to help researchers study the human brain. He spent six years working on various Myst titles including Myst Online as Lead Technical Artist, Technical Game Design, and Content Director. He spent the last few years of his decade in the game industry as the AI programmer and Technical Development Director for Stargate Worlds. In 2009 he began work on a Master's of Divinity degree at Phoenix Seminary with emphasis on Biblical Communication. He welcomes any and all feedback at bslease@gmail.com.

Ready at Dawn

Postcards From a Studio is a new regular column in *Gamesauce*, devoted to profiling studios around the world. We hope to show you some of what makes them tick, what they consider important and the style they want to project. Our first victim, target, profile is Ready at Dawn Studios, located in Irvine California. Ready at Dawn is famous for providing incredibly polished gameplay experiences on the PSP, notably *Daxter* and *God of War*, as well as *Okami* on the Wii.

—Ed

ReadyAtDawn® studios

NAME:
Ready At Dawn Studios

LOCATION:
Irvine, California

YEAR FOUNDED:
2003

NUMBER OF EMPLOYEES:
60 *(5 openings in art & design)*

URL:
www.readyatdawn.com

SOFTOGRAPHY:
Daxter, God of War, Okami

I am met by a large, bodybuilder-type fellow sometimes referred to as "The Arms." More conventionally, he is known as Andrea Pessino, the Vice President of Technology for Ready at Dawn Studios. Andrea seems genuinely uncomfortable with the title (VP of Technology, that is) since he and the other principals at Ready at Dawn would prefer not to have titles at all. But once the company grew to the point where they were hiring for specialized positions, they were obliged to give the new recruits titles in order to make it clear what they were hired to do, and the principals found themselves with titles in the process. As Andrea puts it, "I basically just go do technology stuff where I need to, rather than get wound up over titles. It's there but it doesn't mean anything."

He shows me around a relatively normal studio: cubes in the middle, offices around the edge, toys and posters creating a sense of individuality in each assigned space. I also notice that all of the walls—which you would normally expect to be covered in concept art—have little black curtains over them, almost all of which are closed. Whether it's because they are hiding art or because there is nothing under there, I cannot tell (which is probably the point). In any case, there is apparently enough confidential work going on that they are careful to escort me away from certain areas lest I see something I'm not supposed to. (I'm terribly disappointed, on a personal level, but at the same time gratified to see people taking their confidentiality agreements seriously.)

I am introduced to Didier Malefant, CEO of Ready at Dawn. Didier has an easy way about him that invites casual conversation, but at the same time it's clear he's not interested in mundane BS. I also meet Ru Weerasuriya, the Creative Director at Ready at Dawn. After I spend two minutes in Ru's office, his function is obvious. There are Blizzard-based toys here and there, games all over his shelves, and a large Wacom tablet on his desk. This is clearly the domain of a contributing art and creative director.

> **"We had something to say. Still do in fact. It wasn't dissatisfaction with where we were working—far from it. We just felt we had a way of doing things that would result in quality. When you are owned by someone else, you just don't get the opportunity to be your own boss."**

I notice the furniture and how nicely laid out it is, and Ru mentions that they literally just had this installed. "After six years, we felt it was probably time to just get some real furniture in our offices." Ru and Andrea go on a riff about how different it is from when it was just six guys in an apartment trying to start a business.

Real furniture is just one of the differences from when it was just six guys in an apartment.

We go to a conference room where we get straight to the point of my visit. My first question: “*Why did you guys start Ready at Dawn?*”

Andrea sits back and reflects. “We had something to say,” he says. “Still do in fact. It wasn’t dissatisfaction with where we were working—far from it. We just felt we had a way of doing things that would result in quality. We even got the blessing of those with whom we were working before. We just wanted to do things our own way. When you are owned by someone else, you just don’t get the opportunity to be your own boss. We aren’t there yet in terms of doing our masterpiece, but we are getting there.”

Ready at Dawn deliberately keeps the company small so that everyone knows everyone else, everyone gets input, and they can maintain as much transparency as they can humanly manage.

“We deliberately started small,” he continues, “working on smaller-scope projects to keep from getting out of our depth. Plus, we were just majorly excited about the PSP and what it could mean.” Then he adds: “We don’t have an exit strategy either. So many startups are about people making three or four games, taking the best offer for their studio, and getting out as soon as possible. We deliberately are not going that route. We think we have many games inside of us with this team and just want to keep doing it and having fun. On the flip side though, the moment it’s not fun anymore is when we give it up.”

Andrea explains that they deliberately keep the company small (they currently have about 60 employees) so that everyone knows everyone else, everyone gets input, and they can maintain as much transparency (down to publisher negotiations) as they can humanly manage. They also stick to one project at a time so that everyone is focused on the same thing.

I have to ask: “*How does that scale? What happens when you hit 120 people, or two teams?*”

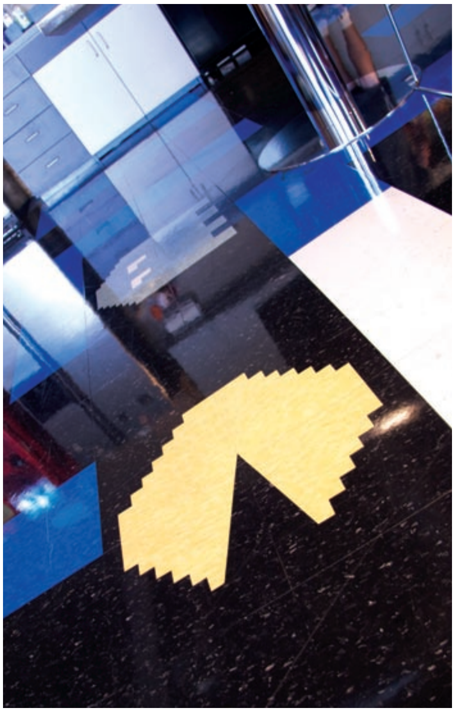
Andrea just shrugs. “Well, if it doesn’t work at that scale,” he says, “then we’ll slim down once again to 60 people.” There is no hesitation, no question about it. The impression is that they have a way of working that works for them—and they don’t want to mess with that equation by growing too fast (or, indeed, *at all* if it means losing that formula). Andrea makes the point that they’ve already been offered large contracts but walked away because they would have required Ready at Dawn to expand too fast.

“*So what is the root philosophy of the studio?*”

Andrea is matter-of-fact: “No Bullshit. If you have a problem with people being honest with you, or you don’t have a thick skin, then Ready at Dawn isn’t the place for you. We *have* to be honest and not dance around issues in order for the quality to be there.” Which leads him to the other part of their philosophy: “Quality is job one. We have no hesitation about holding something back if it’s not up to the quality bar. One of our employees has a great quote: ‘Once the game is on the shelf the only thing you can do is make excuses’—and we just don’t want to. Luckily we’ve had great publishing partners who believe in the same things we do.”

Ready at Dawn’s first game Daxter for the PSP, was consistently ranked in the Top 5 PSP games launched in 2006 with a chart topping Metacritic score of 85.

The other core principle on which Ready at Dawn is based can be found on the company website: making fun games lots of people want to play. What isn’t mentioned on the site is that the *process* has to be fun too. Ready at Dawn has had crunch times in the past like everyone else (*Daxter* is one now-infamous example), but they have learned from those experiences. They now strive for a fun kind of crunch—where people work late because they want to, not because they are told to—the kind of fun that reminds you of why you are a part of the company. *Will there never be crunch* ›



again? Andrea looks distressed. “Well, never say ‘never.’ We do what is necessary to make a great game.”

Ready at Dawn holds many social events—movies, BBQs, holiday events and so on—doing the usual things that smaller companies do to ensure that their employees know each other socially as well as professionally. (While I am there Andrea and Ru even make a play-date for their children to play together.) So it isn’t at all surprising that Ready at Dawn likes to hire people who are fully formed. As Andrea puts it, “We don’t want one-dimensional people. We want people who do things and have hobbies in their own time. Our experience has been that the more two-dimensional people are with outside interests, the more interesting they are as people—and the more effectively they interact with other employees.” They also look for those who will be lifted by the quality that is being produced by everyone else there. “If everyone else is doing great things and you aren’t,” Andrea explains, “we want people who will question why and then bring themselves up to that level.”

> **“Quality is job one. We have no hesitation about holding something back if it’s not up to the quality bar. One of our employees has a great quote: ‘Once the game is on the shelf the only thing you can do is make excuses’—and we just don’t want to.”**

After question time, I go to lunch with Ru, Didier and Andrea. Within moments, abuse and insults are flying across the car—it’s the kind of easy-going rudeness that only people who are comfortable with each other can hurl. And watching the banter at lunch, I get to see that, after working together for six years, after enduring all of the stress that a startup can throw at you, after nearly running out of time and money and closing up shop while building *Daxter*—after all of that, these guys genuinely still like each other and the work they do. And from what some of their employees tell me, that affection extends to the rest of the team as well.

I come away from the visit feeling there is truth to the statement that a studio takes its attitude from the principals. Ready at Dawn is poised for bigger and better things, and I muse how good it is to see independent developers who know where they are going and why.

Ensemble Figures Out How to Go from *Empires* to *Kings*

by Ian Fischer

It was early May, 1998. Nearly all of Ensemble Studio's employees were in San Jose, aboard the Queen Mary. We had just been awarded a pair of Spotlight Awards from the Computer Game Developers' Conference, including "Best of Show." The statuettes looked like miniature Klieg lights, and someone had plugged them in on one of our tables. Sheets and sheets of stickers advertising an entertainment network called Berzerk littered the bar. We had taken to using them to replace the labels on our beer bottles and, as they became more numerous, to toasting "berzerk" and then just randomly yelling "berzerk." Eventually, the people who actually owned the service approached us, cautiously, to ask if we worked for them. By this point, we'd received a number of awards for *Age of Empires* (the game shipped in October of the year before), but this was the first award we'd been given *formally*. Officially. In a ballroom. Surrounded by hundreds of our peers. It was unbelievable. We'd also been informed of our updated sales numbers, well north of a million now, which didn't hurt the mood any. In a few weeks, the entire company would be in Atlanta for E3, attending an award ceremony where we'd take home the Interactive Achievement Award for Best Strategy Game of the Year.

It was an amazing time to be part of Ensemble Studios. But even at this point, while we were still celebrating *Age of Empires*, we were already months into a new struggle—trying to figure out how to do it again.

The man most obsessed with this, our president Tony Goodman, wasn't at the table that night, drinking "Berzerk beer." It would be romantic to claim that he had skipped the celebration to stay back and work, but Ensemble wasn't that kind of shop—Tony wasn't at the table because he had sneaked into the off- limits area of the ship and climbed up one of the Queen Mary's smokestacks.

IN THE BEGINNING

Ensemble Studios had its roots in a consulting firm: Ensemble Corporation. The story of its genesis goes something like this:

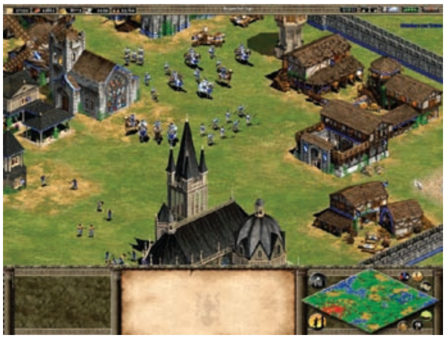
One night in 1993, Tony Goodman was talking to Angelo Laudon, one of the programmers at Corp. While discussing the business, one of them (neither remembers which) asked, "Wouldn't it be more fun to make games?"

Boom. That's it. No multimedia presentations, no spreadsheets, no millions in projected profits for investors, no conversations about "market opportunities." Just two pals who had been thinking about how much fun it would be to make games for a long time.

And that's Ensemble Studios in a nutshell, really. Everything that was good about it, everything that was bad, every success and failure. Right there. "Wouldn't it be more fun to make games?" That was the unspoken vision for the company—take the awesome, manic energy you get when you have two pals working on something they love and scale it up to 25 or 50 or 100 people.

Judging from the end results—no games that weren't in the million-seller club, a franchise that has sold more than 20 million games, a developer that stayed in business for over a decade, a profitable acquisition by Microsoft, exceptionally low turnover—that vision served us incredibly well over the years. But, effective as they may have been in the final analysis, a lot of the principles and practices that defined us also became things that we struggled with.

Barbarossa was a man of great appetites – great ambitions – and a great, red beard. But the question — the question you want to know — is: Was that enough? Is the will of one man enough to forge an empire?



I spent a lot of time playing **Age of Kings** back in the day. My favorite empire was the Chinese and their deadly Chu-Ko-Nu unit. Most days my coworkers and I would forgo food to get a quick round in during lunch time. It was the perfect weight loss plan for me back then!
— Chris Stockman, Blazing Lizard

The problem is that nothing stays still. Once you have worked and worked and worked, and your labor has been rewarded with success, it's tempting to imagine that you have things figured out. That all you really need to do is follow the recipe you perfected the last time and you'll bake the same cake.

This is fiction.

When I think back to the making of *Age of Kings* today, I can't help but do so with that in mind. All of my recollections have a kind of duality to them—these are the things that made us strong… but also made us weak. It was during the development of *Kings* that this duality started to surface—when we first saw that some of the ingredients responsible for making *Age of Empires* a delicious cake would produce very different baked goods in the future.

What follows are six of these "ingredients"—the founding principles of Ensemble Studios that were vital to our initial success and that morphed as we went from start-up *Age of Empires* to follow-up *Age of Kings*.

Principle 1: Full-contact Hiring

If your goal is to put together a company that feels like pals working on something, a good first step is to bring in people who actually *are* friends. Among the earliest members of Ensemble were Rick Goodman (Tony's brother), Brian Sullivan (Tony's boyhood pal), Bruce Shelley (who met Tony at a University of Virginia game club when Bruce was in grad school), Bob Wallace (friend of Tony's father), and John Evanson (son of another of Tony's father's friends).

To achieve and preserve this culture of friends, Ensemble required the entire team to be involved in the decision to bring any new member aboard. Every candidate met with everyone in the company, usually several times. Typically, this meant a lunch or dinner with everyone, a few interview sessions, and a night of playing board games. When these were completed, there would be a meeting to discuss the candidate and then vote. Unanimous assent was required for anyone to get an offer.

The intent was that everyone would own the decision to add someone to the family. And it worked. Nobody ever showed up on Monday to find the new guy "management" had hired, mysteriously in the office. Anyone there was there because all of the others had agreed to it.

New hires knew this too (the full team involvement and time dedicated to hiring was almost universally mentioned as one of the things that impressed candidates about the studio). On their first day, they went in knowing that they had a vote of confidence from all of the people they were now working alongside.

During *Age of Empires*, which required growing the team over a relatively long period of time, this approach to hiring worked well and produced a close, unified crew. *Age of Kings* had a different profile. We needed to hire more people within a shorter period of time. A hiring process that took at least two full days, involved most of the team, and included group dinners after work caused problems as the number of candidates piled up. Some weeks, the average employee would have to block out six or eight hours just for hiring.

Clearly, that was a substantial amount of time, but we weren't worried about any lost productivity—we felt it was worth it to ensure that we hired well. The problem was fatigue. Because of the pace at which we needed to add to the team, the hiring process had gone from being an amazing opportunity to a boring chore for a lot of people. This was but a taste of what was to come—we grew from 20 people to more than 30 between *Empires* and *Kings*, but when we moved on to *Age of Mythology* and beyond, we would cruise past 70 and end in the triple digits. But even during the hiring for *Kings* people were starting to tire of the process.

During *Kings* we also started to see that, although it produced excellent results overall, entrusting everyone with this responsibility could lead to abuse and missed opportunities. For example, someone could think we shouldn't grow past a certain size and then decide to vote "no" on any proposed new hire.

Principle 2: Consensus-based Everything

As the "friends working on something they love" bit might lead you to believe, Ensemble Studios was not a "here are your orders" kind of place. Nothing illustrates this better than the manner in which we decided to actually do *Age of Kings:* the entire company piled into the old conference room, arguing the merits of the various alternatives, then voted on what game we were going to do next. "*Knights and Castles*" beat (if memory serves) "*Elves and Castles.*"

There is a fantastic strength in this approach. The productivity and potential of a group of people working on something they all "see" and want to achieve are magnitudes greater than those of a team that needs specific marching orders for every step. We experienced that strength during the development of *Age of Empires*, when everyone involved seemingly just rowed feverishly in the same direction. We wanted to preserve that feeling for *Age of Kings*.

There are two ways you can achieve this strength. The first is to have a small team. The second is to spend a great deal of time trying to get people to agree. When my business card changed from "Assistant to Lead Designer" to "Lead Designer," I got a crash course in exactly what was involved in the latter. If someone on the team was unhappy about anything (too many hit-points on the towers, silly looking capes on the swordsmen, an ugly font on the UI), it was a problem—and my job largely became dealing with those problems.

Strong as it might be to have everyone in agreement and marching in the same direction, consensus-based operations are difficult to achieve and to scale effectively. As we grew during *Kings*, the process became more and more difficult to manage. We learned that when we hired new people, they likely had no experience with it and certainly could not call upon the countless hours of discussion that had taken place before their arrival. A person who didn't like or play real-time strategy games, for example, would often complain about fundamental elements of real-time strategy games in an attempt to steer things toward the sort of game-play they preferred. (Using logic to explain why you shouldn't attempt to insert a first-person shooter into your already-late real-time strategy game is surprisingly ineffective, by the way.)

Overall, we managed to stick with a consensus-based approach for most of *Age of Kings*. The majority of the systems in the game, from the sheep to the scenarios in the campaign, had their roots in long design meetings that involved the entire company. When we had to make calls without full agreement, we learned another important lesson about consensus-based operations: Once people get used to operating this way, it is very difficult for many of them to accept decisions that aren't made by consensus.

Principle 3: Hardcore Play-testing

During the late '90s, when we talked about our process at conventions and the like, nothing caused more disbelief than the claim that the entire team play-tested. Today it doesn't seem uncommon (most devs I speak with now say their teams do this), but back then it was a pretty foreign concept to have all of your artists stop making art and all of your programmers stop writing code so that they could play their game for a few hours and then give some feedback.

Play-test was the crucible for Ensemble—the arduous gestation, labor, and delivery process through which the game was born. The idea was to get something up and running as soon as possible and then start hammering on it. On the average, at least half of every day I spent running a play-test, taking feedback from the people there (about a quarter of the team), making changes, and sending out summaries and tasks. Additional discussion on anything in the summaries was sometimes taken into a design session, but more often issues were hashed out in giant, endless, multicolored threads that everyone complained about reading.

This play-testing process always worked for us—in fact, the Ensemble alumni at Robot Entertainment and Bonfire Studios still use it today. Looking back, I don't recall any significant signs of problems with this approach to play-testing during the development of either *Empires* or *Kings*. However, where it has indirectly caused some issues—issues that have grown more pronounced over the years—is in the minds of those who were actually there at the beginning.

You see, when you're 22 and you're hell-bent on getting your first game done and you're going to set the world on fire and make the cover of *PC Gamer*, and then you actually *do* get your game done and set the world on fire and make the cover of *PC Gamer*, it burns a set of warm and fuzzy memories into your little sleep-deprived brain—awesome, awesome memories of how fantastic everything once was.

That stuff is great.

Age of Empires and **Age of Kings** singlehandedly caused a 5 year stretch of long lunch breaks and after hours skirmishes (a few times vs. Ion Storm) for a good chunk of the Raven Software team between 1997 and 2003. I can honestly say I've logged more hours playing **Age of Kings: The Conquerors** than any other game in my life. It's easily in my top 5 games of all time.

— Kenn Hoekstra, Pi Studio

Great, that is, until you're working on your next game and the AI has been broken for a week and multiplayer is out-of-synch crashing and you turn to the guy beside you and ask something like, "Remember when we used to all play the game all night every night, and everyone loved everything, and it was perfect, and fans lined up to tell us how awesome and cool we were, and it was sunny all the time, and kittens never died?"

Then it sucks.

Principle 4: Crunching Like You Mean It

Age of Empires had crunch. Not crunch like "we're working until 10"; crunch like "if not asleep, then in office." For months. This was not specifically mandated. There *was* a general call to put in more time—something like: "OK, we need to start crunch next week to get done on time." But to my knowledge, nobody said, "Be here between these hours." People on the team were motivated to put in silly amounts of time.

Kings had crunch as well. It was expected. Some of us looked forward to it. But, unlike crunch during *Age of Empires*, people started to complain about it. Not gripes about the extra hours (which, of course, we had during *Empires*), but real complaints. Complaints of the "I'll have to leave if this doesn't change" variety.

Now, I am a mutant, and I have taken many lumps over the years for this, but: I like crunch.

Yes, yes—it isn't healthy. It isn't good. It's damaging and it should be avoided. I do whatever I can today to ensure that none of the people I work with have to endure it. But my fondest memories, not just of Ensemble, but of life, are of finishing something as the sun comes up, weeks of beard on my face, awake only thanks to gallons of coffee, standing beside others who loved *whatever* enough to do the same. There's some magic there, and I love that with all my heart. I was not alone in this feeling at Ensemble so, for some of us, the degree of serious opposition to crunch that surfaced during *Kings* was disappointing—not because people didn't want to spend dumb amounts of time at work, but because it was a sign that things were changing. With *Empires*, our first time up to bat, there was a universal feeling that we would do "whatever it takes." With *Kings* it felt like we were taking a step back.

There's only so much you can do to change this problem (aside from eliminating crunch). Ensemble already had a general disdain for Mickey Mouse bullshit, which does a lot more than you'd expect to lessen the impact of things like extra hours. Nobody really kept track of vacation or sick days or when someone rolled into or out of the office. Drinks and snacks were free. Lunch and dinner were provided any time crunch was in effect (and you could expense dinner for impromptu late hours any time you wanted). Some of this, like the drinks being free, might seem trivial, but if you don't understand why someone working in your office at 2:00 a.m. is going to be insulted by having to pay a quarter or two for a drink, you probably shouldn't be running a team.

What did change was that crunch went from being unplanned and perpetual to being scheduled and limited. Instead of just announcing a need to start putting in extra time and keeping that in effect until we were done, we started setting hours and picking periods of time—12:00 to 12:00 during the last week of the month, for example. This at least allowed people to plan around crunch and have downtime.

Principle 5: A Cult-like Culture

A lot of people are friends with the people they work with—some more than others. Their closest friends are usually the people they visit around noon, to see about getting lunch together. At Ensemble during *Age of Empires* and the early stages of *Age of Kings*, someone would start marching around the office at noon, bellowing "Lunch Train!" and *the company* would turn out to go get something to eat.

The same thing happened around 7:00 p.m., when someone started walking around recruiting followers for dinner or drinks. On any given Saturday night, a good chunk of the team was at someone's apartment. Friday after work there'd be a dozen people in the play-test area playing *Quake* (then *Quake 2*, then *Half-Life)* until 3:00 a.m. That was the way of things. If you didn't actually share an apartment with the guy you sat next to, you were probably going to be having a beer at his place later that night.

A great deal of this cult-like culture was a function of circumstance. A startup game studio, in Dallas, with no published titles, was not an attractive proposition to most people with experience in the industry. Many of the early members of the team were young, single guys, recently out of college or the Art Institute. Similar age, similar interests, similar situations.

Ensemble's general approach to team-building only reinforced this culture, and not just because of things like the hiring process and the shared hardship of crunch. When GDC and E3 rolled around, we sent the entire team. *The entire team*, as in *everyone*. If we were showing that year, everyone went on to the floor and demoed the game. When the show was done for the day, 20 or 30 geeks clad in shorts and gaming t-shirts loaded onto a bus that, after a short ride, unleashed them upon an unfortunate five-star restaurant. Finished there, the same crew would make its way to a nearby bar to drink until someone did something that got everyone tossed out.

I know that sounds like dot-com-era shenanigans. It wasn't. Keep in mind that everyone went to E3 in 1995, long before *Age of Empires* shipped or Microsoft acquired the studio, when the company was certainly not possessed of vast sums of disposable income. It was part of the philosophy of Ensemble. We sent everyone to demo the game because we couldn't imagine allowing anyone else to show our baby to the world. The company organized and paid for bacchanals at upscale restaurants because we wanted our people to have shared experiences—and the stories to tell that come with those shared experiences.

Of course, in time the culture changed. Of particular note: Somewhere between *Empires* and *Kings* some of us grew up. The guys who could hang out every night in '96 had wives or kids in '98. Through *Kings* and beyond, more and more of our crew fell into the "adult" category, and the balance shifted.

The other thing that changed the cult was hiring. Pre-*Age of Kings*, for better or worse, there was really nobody at the studio for whom Ensemble was "just a job." After *Empires*, we had enough credibility to attract experienced talent, and for some of these people Ensemble was their third or fourth stop. The more we hired, the more we introduced people who had other things going on outside of our circle—people who, at the end of a day of demoing a game, wanted to go back to their hotel rooms.

As with complaints about crunch, this was difficult for some of us to understand or deal with. It's not hard on an intellectual level to comprehend why any person who just endured six hours of flashing lights and looping sirens would want to enjoy some quiet for a bit. Still, we just did a fantastic job of selling a game for which we all worked our asses off, and now we're all going out to celebrate together… and someone doesn't want to go?

That change is not the kind of thing you can "fix." You cannot say, "Everyone must go out and have fun together." But when it doesn't happen, there's a sense that something is amiss.

Principle 6: Free-range Development

I have a form of narcolepsy that sometimes visits me with acute insomnia. One night, my eyes pop open at 3:00 a.m. and I know I'm not going to fall back asleep. I get dressed and head into the office. I'm off the elevator and headed for my desk when I hear something a few offices down, so I detour to check it out. Angelo Laudon is there. Writing code for *Age of Kings*. He couldn't sleep either.

There's a chapter in Steven Pressfield's *The War of Art* that, paraphrased, states that you're doing the right thing if you'd do it even if you were the only person on the planet. Arnold Schwarzenegger, he suggests, would still go to the gym even if there was nobody around to impress with his physique. Likewise, I'm certain Angelo would still code games were he alone on Earth. In those days, Ensemble was powered by people like that—people doing, in essence, "this thing they were born to do."

That isn't to suggest that there wasn't a massive amount of planning and scheduling. Guys like Rick Goodman (lead designer on *Age of Empires*), David Pottinger (lead programmer), Brad Crow (art lead), Chris Rippy (producer), and Harter Ryan (executive producer) certainly did a great deal of work to organize and direct development. Still, everyone had a great deal of freedom and might head down one path or another.

Frequently, this creative freedom had fantastic results. Numerous pieces of *Age of Empires* were completed because someone thought a feature should be in there and felt like working on it. It was not unusual to roll into the office on Monday and find an "over the weekend" email informing everyone that some horrible bug was fixed

Age of Empires II came out while I was in high school and it taught me some great history lessons. After enjoying the Barbarossa campaign, I learned about his failure to reach Jerusalem in one piece (he got there pickled in a barrel). In history class later that week our teacher was covering Hitler's invasion of the Soviet Union in World War II. As soon as he said Hitler called it Operation Barbarossa I blurted out "No wonder it failed!" and then had to explain how I was learning stuff from video games. What a great game.
— James Everett, Sidhe Interactive

simply because someone had grown annoyed with it. The artists learned tricks on *Age of Kings* that allowed them to do things with a 256-color palette that nobody thought they should be able to do, not because it was planned, but because they decided to start playing with various techniques.

Empires, despite being Ensemble's first game, did have an advantage in that it had time to find its way a bit more than *Age of Kings*, which Microsoft initially wanted delivered after only one year of development. The limitations of a 12-month schedule did not pair well with established freedom to pursue objectives you found interesting.

Nowhere was this more obvious than in the initial design of the game. While envisioned at the start as simply moving *Age of Empires* into the Middle Ages, the initial design direction for the game consisted of mixing in every idea that hadn't made it into *Age of Empires*. The poster child for this was a full-screen diplomacy matrix which allowed players to negotiate all manner of convoluted relationships. However, there are dozens of other things we worked on that did not pan out, including unit facings in combat (micro each of your 200 little guys to stab enemies in the back), off-map trade (sometimes you send a trade ship out and it just doesn't come back—you don't know why), and pillaging (if you killed a villager who was gathering, he'd drop a lump of whatever he had on him). We also worked on persistent fire, mercenaries, outlaws, roads, heraldry design and display, raider civilizations, renewable resources—all interesting ideas that didn't make it into the final game.

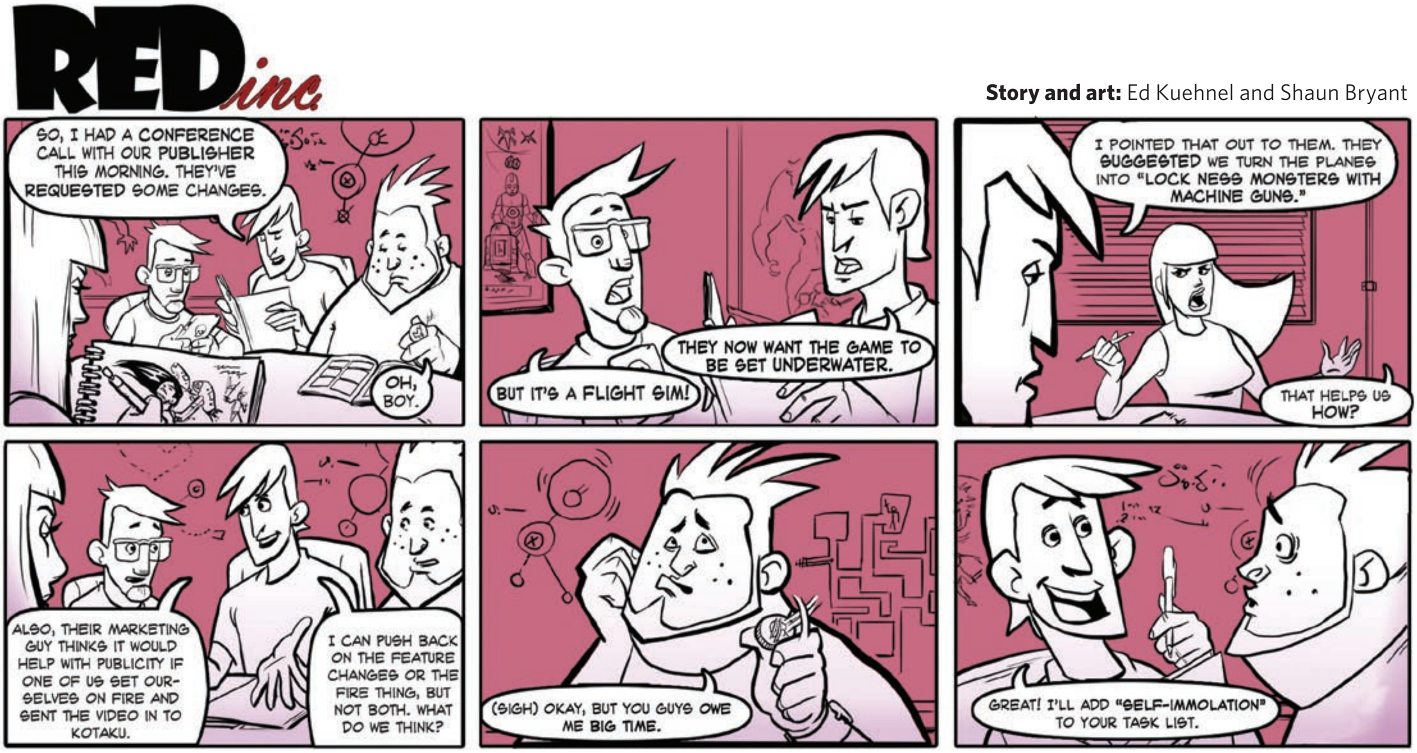
The freedom to pursue things like this was a double-edged sword, and it forced us to reset the design of the game and spend an additional year on development. It would also bite us later in development when people decided to rewrite systems like the AI and pather instead of improving what had shipped in *Empires*. Consequently, we had to pull people off our second game and engine team to finish *Kings* on time.

Here We Go Again…

It's late summer, 1999. We have just learned that the last release candidate we put up has been accepted and *Age of Kings* is headed for manufacturing. Thirty-some people are popping bottles of Dom Perignon in the cafe of our new offices.

This is just one more thing that has changed. Two years prior, a handful of guys crawled out of an un-air-conditioned ex-dentist's office to celebrate finishing *Age of Empires* with Moons Over My Hammys. Now we're drinking $200 bottles of Dom on the second floor of an office we had built out to look like the Star Trek ride in Las Vegas.

In a few hours, we'll descend upon The Bavarian, the traditional final stop for any Ensemble project—a restaurant we've terrorized so thoroughly that we must pretend to be with Texas Instruments when making the reservation. Which is to say, even in times of change, some things stay the same.


RED inc.
SO, I HAD A CONFERENCE CALL WITH OUR PUBLISHER THIS MORNING. THEY'VE REQUESTED SOME CHANGES.
OH, BOY.
THEY NOW WANT THE GAME TO BE SET UNDERWATER.
BUT IT'S A FLIGHT SIM!
I POINTED THAT OUT TO THEM. THEY SUGGESTED WE TURN THE PLANES INTO "LOCK NESS MONSTERS WITH MACHINE GUNS."
THAT HELPS US HOW?
ALSO, THEIR MARKETING GUY THINKS IT WOULD HELP WITH PUBLICITY IF ONE OF US SET OURSELVES ON FIRE AND SENT THE VIDEO IN TO KOTAKU.
I CAN PUSH BACK ON THE FEATURE CHANGES OR THE FIRE THING, BUT NOT BOTH. WHAT DO WE THINK?
(SIGH) OKAY, BUT YOU GUYS OWE ME BIG TIME.
GREAT! I'LL ADD "SELF-IMMOLATION" TO YOUR TASK LIST.

Story and art: Ed Kuehnel and Shaun Bryant

A Conversation with Randy Pitchford

1. Some background on Randy Pitchford? Who are you and what do you do all day?

I am the President of Gearbox Software. I spend most of my time collaborating with the incredible game makers, businessmen, entertainers and technologists at the studio to help us develop our priorities and to create the partnerships and allocate the resources necessary to mobilize activity towards those priorities. I tend also to commit myself to creative development as much as possible as my interests in entertainment and the craft of game-making are why I became so dedicated to this industry. And, of course, I spend considerable amounts of my own free time playing games of all kinds, and this passion pre-dates any professional involvement I had with the craft.

Harstock in the Hallway from Gearbox's BIAHH

2. Tell us about Gearbox Software. What led to its formation?

Several years earlier, Brian Martel and I met while working together at 3D Realms, where we had the opportunity to contribute to *Duke Nukem*. After helping to ship the *Duke Nukem 3D: Atomic Edition* and the shareware version of *Shadow Warrior*, we parted ways with the studio and began to branch out on our own. At an interim start-up we had the opportunity to meet and work with the other founders of Gearbox.

Brian Martel, Stephen Bahl, Landon Montgomery, Rob Heironimous and I founded Gearbox Software in the beginning of 1999 and completed our first game, *Half-Life: Opposing Force*, later that year. It was during that time that we discovered the principles and philosophies that would guide us as we planted the seeds that have grown into the studio as it is today.

3. There are rumblings that you were a magician at one point. How does performing magic compare with making games?

It's true: I used to be a professional magician. I worked at a magic-themed night club in Hollywood called Wizardz and occasionally would appear at the famous Magic Castle. I was married at the Magic Castle, actually—it was quite a memorable experience!

Magic, like many video games, is a form of entertainment in which we strive to help our audience succumb to fantasy and enjoy experiences that are impossible in the real world. We use similar tactics, such as storytelling and misdirection, to help guide our audiences and encourage them to trust us and come with us for an adventure in which they will experience wonder and joy—that is, if we do our jobs well.

With magic, I am able to quickly develop a personal relationship with the audience and earn immediate feedback regarding the quality and impact of the entertainment I'm providing them. In video games, the relationship with the audience is more remote and more distant. The trade off, of course, is that with video games we can entertain millions of people. The size of this audience is very exciting to me—it makes our collective effort more meaningful and relevant.

4. Is Gearbox your first entrepreneurial venture?

Gearbox Software isn't my first company. Now that I think about it, it isn't my third, either. I had my hands in a number of things earlier in life. One of my earliest and best lessons in leadership, management and corporate organization occurred when a friend and I were window-washers together and decided that we could take what we knew about the window-washing business and scale it up. We created a franchise-style business that involved dividing Los Angeles County into territories, and for each territory recruiting a small army of managers whom we trained and supplied with equipment. In our first summer, we quickly grew to 55 employees and were very likely the largest residential window-washing company in Southern California.

When I moved to Dallas, I gave my interest in the company to my partner, and from what I understand, he was able to do quite well with it for many years. I think he eventually sold the business, and I believe he's an investment banker now, but it was a fun thing to do and a great learning experience while we were in school.

5. Where did **Borderlands** come from? What's in the future for that franchise?

Borderlands came from Gearbox Software! Years ago, we believed that there was a lot of fun to be had if we married the discovery, growth and choice that come from loot-driven RPG's with the intense, moment-to-moment fun of an FPS. The effort to figure out how to blend these genres in a way that would be fun for us and our customers took many years—and *Borderlands* is the result. Today, we can see how *Borderlands* performed when tested in the market, and we are very excited about what we've learned. Because of the nearly five years we invested in developing the technology and learning, the future of what we do with the brand and the game design is very exciting to us. Currently, fans of *Borderlands* will find that we are rapidly investing in adding value to the game through downloadable content. We've released two DLC packs for *Borderlands* on all three of our platforms (Xbox, PlayStation and PC) and have another coming soon.

> "The effort to figure out how to blend these genres in a way that would be fun for us and our customers took many years—and Borderlands is the result."

6. Who do you look up to and why?

My heroes are people I know very well—people I work with every day. So many of my heroes are at the studio that naming them specifically—and explaining why they are my heroes—would be far beyond the scope of this interview—and probably a little too embarrassing for those people.

Screen capture from Gearbox's Borderlands

7. What happened with the **Aliens** game? Will this see the light of day?

We love *Aliens* and care very much about the brand and the game within it. I hope we won't have much more of a wait before we can share more of our efforts there. What we've done with the game is very exciting.

8. There have been some out-of-context quotes from you floating around recently. What happened?

It's always interesting to me when video game "journalists" earn value and readers from contrived controversy. Somehow, they've managed to leave their readers with the impression that some of my favorite game-makers and some of my favorite games are things I wish to tear down. Only from the greatest do we learn the greatest lessons. Sometimes, when I'm talking "shop" with developers or journalists, it's fun to explore the industry a bit and look at the greatest and imagine what could make the greatest even better. It's comments made there that can get taken out of the context of a larger discussion and turned into what appears to be outright criticism or even attack. It's shameful, but it doesn't really matter in the grand scheme. I'm not really important. What's important is that this industry keeps driving ahead to reforge the definition of entertainment and how it impacts and reflects culture and art and joy on this planet.

9. So what kind of car are you driving these days? In "The Perfect Randy World," what would you be driving?

I drive a British sports car. It's not pretentious and certainly not ostentatious. I am a fan of performance, but I tend to make transportation choices that are about how the driver thinks and feels—not about what the guy watching the driver thinks and feels. I'm very happy with my current car and hope that when it's time to switch, something as suitable is on the market.

Hallway combat screen capture from Gearbox's Aliens Colonial Marines

10. What are your thoughts concerning emergent game-play versus scripted moments?

Why must these be mutually exclusive? Much fun can be had playing with systems that are very dynamic and reactive. Meanwhile, a crafted moment can very effectively create joy or suspense, drama or excitement. Interactive entertainment can and should flexibly use any tool at its disposal to create gratification and fulfillment and a sense of accomplishment within our audiences. We game-makers exist to bring entertainment to our customers. Any viable means to that end are fair game, so to speak.

come join the zynga family!

now hiring:

web developers, back end developers, flash engineers, & product managers

contact jobs@zynga.com or visit www.zynga.com for more info. follow us on twitter too, @zynga!

Cancellation Notice

WHAT TO DO WHEN YOUR GAME GETS THE AXE

It almost always sucks. There you are, flogging your guts out on a project that is progressing when all of a sudden the word comes down: it's canceled. Everyone is milling around in confusion. What happens next? Will we get let go? What the hell happened? The letdown is palpable, and the lack of knowledge of what actually took place is crushing. Rumors fly, the blame game and finger-pointing intensify. You hear (and utter) dark words in dark booths at the back of dark bars. Before long, it's "them against us," developers turning against those who did the canceling.

Then again, sometimes it doesn't suck. Sometimes you have no confidence in what you're making. Sometimes you have chopped and changed so much of what the game is supposed to be that you just want to do something—*anything*—different, something in which the course is charted and more defined. And sometimes you have been on a death march for months, working late and through weekends on a morale-sapping project from Hell. In these cases, getting canceled is more like a celebration.

Even so, from a business standpoint, getting canceled is generally bad news—a threat to job security and the viability of the enterprise. And since it could happen to any of us at any time—tomorrow for example—it's better to be prepared. So the question we have to answer here is: When you get canceled, what *do* you do next?

FIRST THINGS FIRST

Well, the first thing you do is go out for a beer and get the bitching and blaming out of the way. But be careful where you point fingers. This is a small, incestuous little industry, and the wrong bitter word directed at the wrong person at the wrong time can haunt your career for years. If people get laid off, don't point the fingers at them, and if you are among the layoffs, don't go on a bitter tirade. It might be satisfying in the short term, but it'll come back to bite you in the future. On the other hand, some air clearing isn't the worst thing for your sanity either—depending on who you do it with and how it's done.

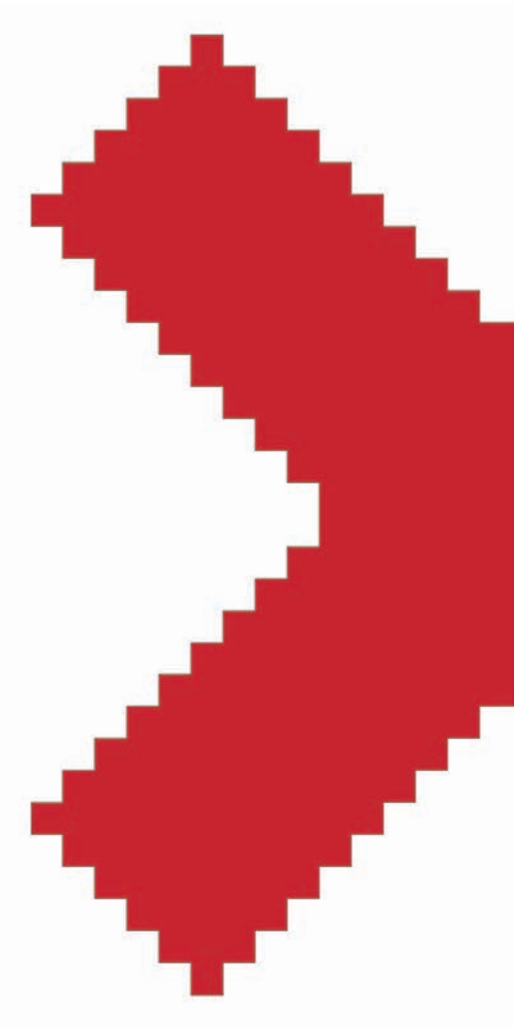
STAY BUSY

The next thing to do is *find something to do at work*. There is nothing worse than coming in the next day and just playing solitaire all day because no one has told you what to do. The more you sit around aimlessly, perhaps thinking that you're just punishing those above you for a bad decision, the more likely you are to become the target of a layoff. Rather than making a crappy situation worse, stay busy and use the time to your advantage. Learn some new software, or do concept art for a new game, or work on a small prototype. More than one great game has come out of people finally having the time to build a prototype at work. And of course, don't forget to update your résumé and gather samples of what you've been working on, in case the team is laid off.

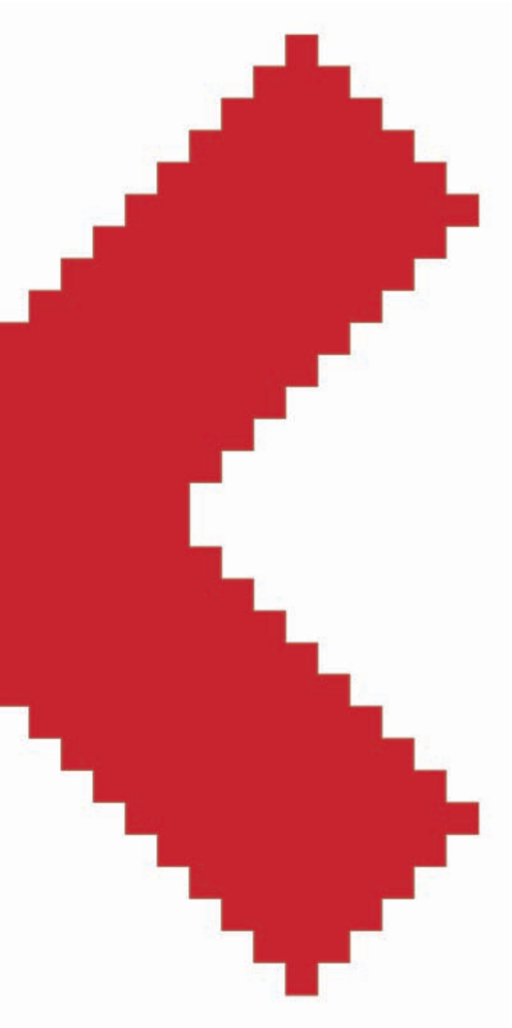
Rather than making a crappy situation worse, stay busy and use the time to your advantage. Learn some new software, or do concept art for a new game, or work on a small prototype. More than one great game has come out of people finally having the time to build a prototype at work.

PERSPECTIVE

While all this is going on, it's vitally important to *have some perspective on the whole situation*. That may be easier said than done, especially when you've been down in the trenches, overloaded with tasks, suffering the impact of brain-dead decisions handed down from on high, better able than management to judge what's going on and know what to do to fix it. In spite of all that, you must attempt to put yourself in the position



of those making the decision. Think about the circumstances that would lead *you* to cancel the project and potentially put people out of work.



There was a famous case recently in which a company canceled an ongoing game that had been in development for years, putting a team of 30 out of work. Why on earth would they do that? To put it simply: The team ran out of money. Although they went to the publisher to ask for enough to finish the game, the publisher responded by offering a deal so fraught with developer risk that in actual fact the incentive to finish at all was removed. If you're an indie developer, and you work your butt off to finish a game only to lose all rights to the IP (and any chance of sharing in the profits it generates), then what exactly are you working for? That's especially true if you're one of the guys at the top taking all the real risk—one of the ones with a second mortgage on the line. You lose the promise of a back-end payout and suddenly it's no longer worth the worry and responsibility to keep everyone else employed.

The people at the top don't make the decision to cancel a project in a vacuum. In fact, you can generally assume that before making that difficult decision they have suffered many sleepless nights worrying about what's going to happen to their people.

There are many, many things that can happen for those in charge of development to suddenly decide the effort is not worth the result: getting squeezed by the publisher who always seems to want more for less; the publisher suddenly wanting to renegotiate the contract on more favorable terms; lots of milestone submission kickbacks, with corresponding delays on milestone pay-outs. The list of publisher tricks for driving a developer to desperation is endless. However, of this you can be sure: The people at the top don't make the decision to cancel a project in a vacuum. In fact, you can generally assume that before making that difficult decision they have suffered many sleepless nights worrying about what's going to happen to their people. While that may not be of much comfort to you when months or even years of effort are yanked out from under you, it's worth understanding that those in charge aren't the enemy and they aren't enjoying this either.

HOW A PUBLISHER SEES IT

At the publisher level, there can be many reasons for canceling a game—some legitimate and some less so. For example, the publisher may perceive a lack of progress or become frustrated by a developer's inflexibility—either of which might simply come down to the basic relationship between the publisher's producer and the developer. Typically, at that point there is a "council of war" at the publisher during which is determined what has been spent to date, how much will have to be spent to push the game out the door (a figure that is often a *lot* less than the developer would like it to be), and how much potential earnings are possible from the game if development proceeds.

The publisher also must determine whether the money which could be made is worth the backlash of having a crap game in their catalog. No doubt, the question will be whether it's better to try to recoup some of the investment through sales or to simply write off the money spent to date and flush the game entirely. Often when a license is in play, the publisher has to pay for it whether or not the game is shipped, in which case *whatever* the developer produces is shipped, regardless of quality—anything to get something back on the cost of the license. No doubt this is why EA pushed out *Superman and Cat Woman* even though they were very aware of the shortcomings of the actual implementation. Better to earn something than nothing.

WEIGHING OPTIONS

However, making the choice to bring a lousy game to market just to make a few bucks must be balanced against the costs of doing so. There are the obvious hard costs, including QA, marketing, internationalization, and continuing developer overhead. But there are also some soft costs, such as having a black mark against you in the minds of players and reviewers. Rather than face those unsavory alternatives, sometimes publishers will just cut the risk and move on, not wishing to send more good money after bad.

Developers carry some of the responsibility for cancellations too. Quite often developers will promise the world, defining an outrageous scope just to get the gig.

Those games that get cut tend to fall into one of three categories:

- Games with insufficient quality to match the established standard in a familiar genre (FPS, Racing, etc.)
- Games which are over-budget and under-quality with no apparent end in sight
- Genre-bursting games with inherently unpredictable prospects

Category three is the one that rightly pisses off a lot of developers. Publishers are ruled by sales forecasts, and if their forecasters aren't able to give somewhat firm predictions—because a genre or game mechanic is so new that they can't find a parallel—the publisher makes its decisions based on the data on hand—which is to say based on *what it costs to make the game.* Unless they are the shit from day one, revolutionary games tend to represent the highest risk a publisher can take, which is why they are the first to go when costs start spiraling.

ON THE OTHER HAND

That's all giving publishers the benefit of the doubt. On occasion, some publishers also engage in conduct that can't be called anything but manipulative. There are times when a publisher knows of the knife-edge a developer is on, and then uses milestone delays to edge that developer towards bankruptcy. At the last minute they will offer to renegotiate the contract, leaving the developer with Hobson's Choice: either give up IP rights (and the real value over time) or go bankrupt. In extreme cases the publisher may even push the developer into foreclosure, buy up the IP rights and development assets at auction, then offer the original development team contracts to finish the game they were developing and used to own! (This really happened to a developer in the UK a few years back. Thankfully that publisher is now out of business too.) Sometimes a publisher will cancel a perfectly good game simply because it doesn't fit its portfolio, or because it was the pet project of a bigwig who has since been replaced by new management.

At the same time, developers carry some of the responsibility for cancellations too. Quite often developers will promise the world, defining an outrageous scope just to get the gig. (Inexperienced developers are especially prone to underestimating costs and time in this way.) Developers with multiple teams often set themselves up for cancellation when they take money for one project but end up spending it on another which has run out of funding, or when they promise the A team for a specific project but switch to the B team once funding has been secured.

SWEET SWEET OPTIMISM

Developers are, almost by definition, very optimistic about scope and their ability to get things done—programmers in particular. Often they genuinely believe that they can do what they propose, but too often they fail to factor in sufficient this-didn't-go-as-planned time. Another common mistake: Developers don't produce visual improvement with every milestone, making it impossible for the publisher to see any tangible progress.

As things progress and it becomes apparent that the project isn't holding to the promised schedule, a rift may open up between the publisher—who sees costs without progress—and the developer—who says: "It's all in hand. We just need a little more time… or money… or understanding."

TWO POINTS OF VIEW

This is the essential problem of publisher/developer relations and what results in so many games being canceled: Developers

Mochi Media fuels the creativity of Flash game developers by providing them with tools to monetize, distribute and track their games. We are the world's largest online games property reaching over 100 million unique users each month.

- **Monetize** your games through advertising and microtransactions.
- **Distribute** your game across Mochi Media's network of 30,000 publishers in our partner network.
- **Engage** your gamers with easy Facebook integrated challenges and gamer profiles.

PARTNER WITH US!

Seeking Game Developers For Premium Flash Games!
To learn more visit

www.mochimedia.com/coins

PARTNER WITH US! WWW.MOCHIMEDIA.COM/COINS

This is the essential problem of publisher/developer relations and what results in so many games being canceled: Developers are about "the game," and publishers are about "the money."

are about “the game,” and publishers are about “the money.” Developers are artists who regard money as a means to an end, whereas publishers are often publicly-owned entities that have to worry about quarterly sales figures and keeping stockowners happy. At root, it’s the basic art vs. business dichotomy. Developers make games, while publishers make money. It’s a small but absolutely crucial difference and one that every developer (and publisher) should have tattooed on their foreheads.

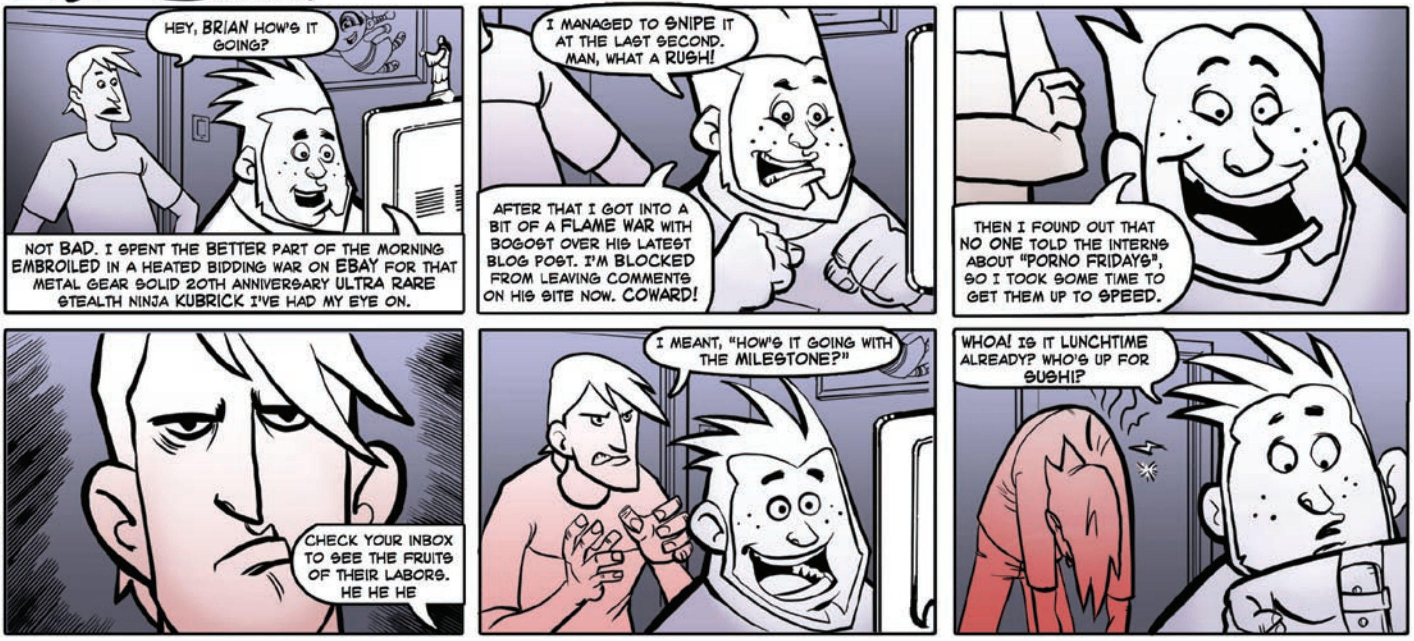
Boiled down, cancellation is a reflection of the relationship between a developer and a publisher. The better the relationship, the less chance there is of cancellation (assuming, of course, that the game itself has reasonable merit). Understanding that, however, does not make your situation easier when your baby has been killed by someone else who should have seen how great it was.

In the end, perhaps it’s best to simply treat a cancellation as a learning experience. One of the principals of the failed developer Rebel Boat Rockers once told me that he learned far more through failure than he ever did through success—and considering that he has since gone on to head up a highly successful company, there must be some truth to it. Those who succeed right out of the gate often aren’t sure how they pulled it off, whereas those who fail before succeeding are in a much better position to repeat those things that work and avoid those that do not.

And besides, canceled ideas can often resurface in other, better games. If nothing else, you might think of cancellation as a chance to put good ideas on ice until you can make better use of them.

RED inc

Story and art: Ed Kuehnel and Shaun Bryant


HEY, BRIAN HOW'S IT GOING?
NOT BAD. I SPENT THE BETTER PART OF THE MORNING EMBROILED IN A HEATED BIDDING WAR ON EBAY FOR THAT METAL GEAR SOLID 20TH ANNIVERSARY ULTRA RARE STEALTH NINJA KUBRICK I'VE HAD MY EYE ON.
I MANAGED TO SNIPE IT AT THE LAST SECOND. MAN, WHAT A RUSH!
AFTER THAT I GOT INTO A BIT OF A FLAME WAR WITH BOGOST OVER HIS LATEST BLOG POST. I'M BLOCKED FROM LEAVING COMMENTS ON HIS SITE NOW. COWARD!
THEN I FOUND OUT THAT NO ONE TOLD THE INTERNS ABOUT "PORNO FRIDAYS", SO I TOOK SOME TIME TO GET THEM UP TO SPEED.
CHECK YOUR INBOX TO SEE THE FRUITS OF THEIR LABORS. HE HE HE
I MEANT, "HOW'S IT GOING WITH THE MILESTONE?"
WHOA! IS IT LUNCHTIME ALREADY? WHO'S UP FOR SUSHI?

Great People. Big Ideas. Still Growing.

To get in touch with our partner relations team, contact
gamesubmissions@bigfishgames.com

CONFLICT AND CALL

Resolving Your Principles in the Video Game Industry

By Bill Slease

In my decade or so as a paid game developer, I've often been asked two different questions by people inside and outside the industry. The insiders want to know how I can justify making video games for a living given the beliefs I profess, and the outsiders want to know why on Earth I would want to make games in the first place. At the risk of speaking for other developers with strongly held principles, the answer to both questions starts with the same observation. The obvious conflicts you're thinking of aren't the *really* problematic ones. In other words, *the situation is actually worse than you think*.

With non-gamers, I have to do some de-mythologizing. *Gamesauce* readers don't need to be convinced that:

- Games don't turn people into social misfits.
- Gamers come in all ages, genders, races, shapes and sizes.
- Gaming is a social activity.
- *DOOM* is as responsible for school shootings as *Pac-Man* is for obesity.

(An aside: If you find yourself needing to lay any of this groundwork yourself, John C. Beck and Mitchell Wade have written a competent book full of encouraging data and analysis called *Got Game: How the Gamer Generation is Reshaping Business Forever.*)

There are, however, two common critiques of video games in general that have some validity: that they are prone to gratuitous violence, and that they objectify women. These critiques illustrate two areas that create conflicts for me—let's call them "Conflicts of Attitude" and "Conflicts of Intent." Although these are the most obvious sources of potential conflict, there is a third category that for me is even more troublesome as a Christian: I'll call this last one "Conflicts of Content."

CONFLICTS OF ATTITUDE

While I firmly deny that video games cause the violent crimes we see today, all you have to do is take your grandmother to see *District 9* to convince yourself that we've been desensitized to violence. While you are getting a kick out of the obvious *Half-Life* references, watch Grandma's reaction to the gibbing that is barely even registering in your brain pan. Our consciences have been seared my friend, and we can no longer deny it.

The major issue here is the amusement we experience at the expense of the sanctity of life. Is this a new problem our generation has foisted on the world? Not at all. Ancient Roman gladiatorial arenas were far worse—but we're not going to score many points in eternity by being better than the Romans.

Regardless of whether the general trajectory of humanity's morality is headed up or down, it would be hard to argue that shooting avatars in a video game has anything in common with pulling a real trigger on a real person—except when it comes to the attitude of the heart. Rand Miller, creator of the *Myst* series, once described to me the sense of loss he felt when he watched his daughter play *Tomb Raider*. When she began playing, she was distraught over having to shoot a dog to progress. Sadly it didn't take long for her to overcome her reluctance.

Rand's daughter is lucky though. She had her father nearby to point out the change—and the lesson. Too often young gamers are left to fend for themselves in situations at least as complex as any competitive team sport. In traditional sports, coaches teach young players how to not let the game get the best of them. Antisocial behavior is quickly corrected by a mentor who can teach the player to be not only a better player, but a better person. On the other hand, in gaming we often see young players left on their own to assert themselves however they see fit, with terrible consequences to personal character.

The obvious conflicts you're thinking of aren't the *really* problematic ones. In other words, *the situation is actually worse than you think*.

Can the game developer solve these problems? Not all of them. Are there at least some things we can do to help? Is there a way to preserve the game mechanic that player death affords while also honoring the sanctity of life? Is it possible to make a game with enough routes for progression that I don't force a moral conflict on the player (or the developer, for that matter)?

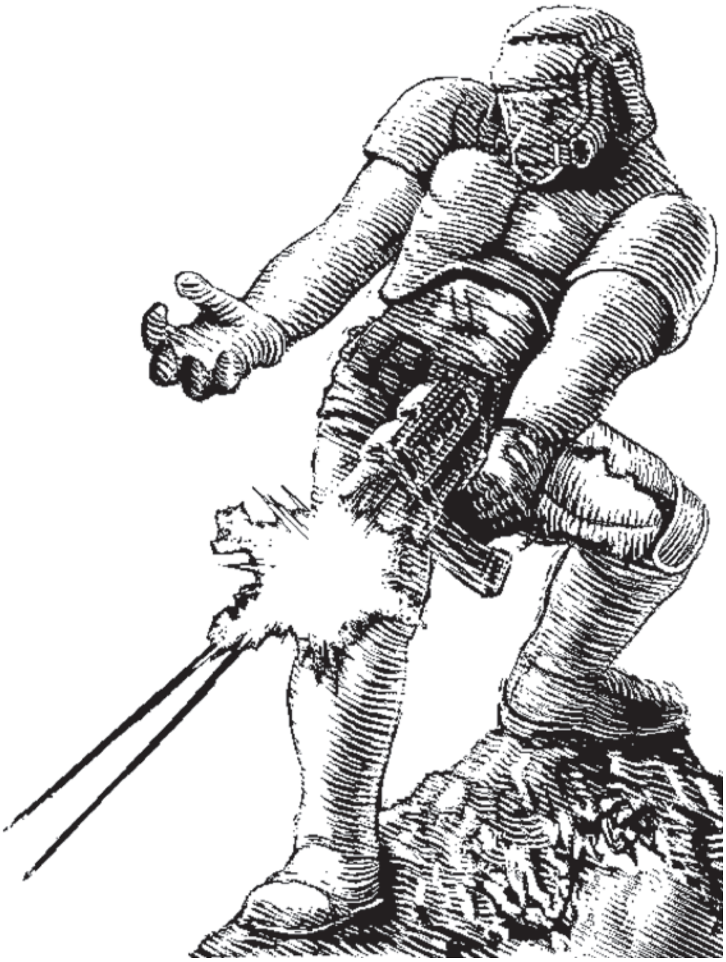
CONFLICTS OF INTENT

It's usually worth a laugh when I tell people that video games aren't going to stop objectifying women until either Jesus returns or all the men die. If it makes women feel any better, men objectify everything, not just women. And yes, based on the Biblical message that the creation of every single human being is an intimate and intentional act of God—that all humans are beloved of God and gifted with a unique responsibility and position in the created order—it's a problem.

Here's the kind of torture I put myself through: Suppose my boss decides our characters aren't "realistic" enough, so he wants me to work on a jiggle algorithm (you know what I'm talking about). And just like that, I feel conflicted. I think to myself, *I'm just trying to simulate reality. There's nothing wrong with it—God invented the jiggle after all*. But then I also think to myself, *Jesus said that if we even look at a woman with lustful intent, we've committed adultery in our hearts. I don't think he would want me to encourage others to be lustful.*

Here's the kind of torture I put myself through: Suppose my boss decides our characters aren't "realistic" enough, so he wants me to work on a jiggle algorithm (you know what I'm talking about). And just like that, I feel conflicted.

The difference between God putting jiggle in the world and us putting jiggle in our game is authorial intent. God's purpose in creating jiggle is divine: To encourage men and women to "be fruitful and multiply." God's intent of the feature, if you will, is to make obeying him more pleasurable. Our purpose, on the other hand, is probably to titillate gamers and make more money. I'll spare you a bunch of theology here and simply state it this way: God has a noble purpose. As for us? Not so much.



Original Doom cover art by *Don Ivan Punchatz*, Reinterpretation by *Jim Moore*

CONFLICTS OF CONTENT

There are conflicts that have caused me deeper trouble, and these are of a different sort. They have to do with games (and game developers) as teachers.

I highly recommend James Paul Gee's book, *What Video Games Have to Teach Us about Learning and Literacy.* In it, Gee makes a thorough argument for games as teachers and explores the kind of teaching they do. It turns out games are really, really good teachers. The scary part for me is that I know very few game developers who are consciously considering what their life's work teaches others. In particular, I'm concerned about the way in which games teach and influence a player's world-view.

Think about it this way: Very few people can articulate a philosophy by which they live and act, but everyone has a basic way in which they view the world. We have answers to a set of core questions, whether we can articulate those answers or not. Our answers to these questions drive how we make decisions and how we behave. These questions include:

> Who am I?
> Where am I?
> What's wrong with the world?
> How can it be fixed?

Let me give you an example of how my religious world-view can sometimes create a sense of conflict in my profession: For most of my career, I have had the good fortune to work on projects that reflect a world-view I could affirm as a Christian. On my last project, however, I did not agree with the game's world-view at all, and that disagreement created significant internal conflict for me. The product depicted a godless universe in which man's purpose is to discover within himself the knowledge or understanding (it's not clearly specified) that will allow him to evolve to a higher state of being. This is essentially the opposite of what the Bible teaches about who we are, what's wrong, and how it can be fixed. I was in a position to suggest a story-line that would inject an alternative view, but another direction was chosen. This issue remained an internal conflict for me for my entire time on the project. I felt as though my participation in the project put me in a position of teaching others (through the game) an idea which is not only inherently false but potentially dangerous (theologically, anyway). As a Christian, how could I go to work every day and put my thumbprint on a project that encouraged others to put their faith in man instead of God?

WHY BOTHER?

These conflicts don't have nice and pretty resolutions. But they do bring me back to the question people outside the industry ask—the one you are probably asking as well: Why bother?

I bother because I think God cares about video games just as he cares about all of humanity's work. There are those who believe recreation and entertainment are wastes of time. But I see described in the Bible a God who not only cares about these things but ordains them, and I see no reason to exclude video games.

I also bother because my work is part of a larger search for meaning. Theologians use the term *normative* to describe the state of

things as they were intended by God. Part of my work as a thinking Christian and a game developer is to discover what's normative for video games specifically and recreation in general. This is a task-in-progress, but I've stumbled across some intriguing clues. Here are three clues then as to what I think is normative, or in other words, what God's intent is for video games:

Clue #1: Games Are Teachers

For sheer practical value, we can learn a great deal from how and what video games teach. The problem teachers encounter when trying to integrate video games with existing curriculum is that they can't figure out how to assess what the game is teaching. They want to be able to have students play a game like *Civilization* and then answer a multiple choice test about the progress of history. As a result, they miss what games are doing, and that the game is already assessing what it's teaching. You progress in games by learning what they want you to learn and applying it in situ. Giving someone a facts assessment after conquering the known world is like asking Michael Jordan to sit down and draw a detailed, to-scale map of a basketball court. He might be able to do it, but it rather misses the point.

Games teach tacit skills and give players experiences from which they can derive general principles that otherwise they would only have learned years later, at great risk and expense. Granted the experiences are based on a simplified version of reality, but so are most things educational. Games also provide what Gee calls a true psychosocial moratorium—a space where it's safe to fail. This kind of environment is essential to learning, and it's important to note that schools in general fail to provide this. Video games are the only place I know where people can really relax and play, experiment and learn relatively consequence-free.

A decent working definition of wisdom is the capability of applying knowledge well. This capability generally only comes with experience. Dare I say, games can impart wisdom? How much more important it is, then, for us as developers to consider what our games are teaching. We have the power to shape a generation experientially. Uncle Ben told Peter Parker (Spiderman), "With great power comes great responsibility." He was paraphrasing Jesus, who told the Apostle Peter, "From everyone who has been given much, much will be required" (Luke 12:48).

My work is part of a larger search for meaning. Theologians use the term *normative* to describe the state of things as they were intended by God. Part of my work as a thinking Christian and a game developer is to discover what's normative for video games specifically and recreation in general.

I'm convinced that teaching wisdom is a normative purpose of games. If that's true, we need to be able to figure out and articulate *what* exactly is being taught.

Clue #2: Games Are Verbs

An interesting way to begin to identify what a game is teaching is suggested by Raph Koster in his book, *A Theory of Fun for Game Design*. He contrasts video games with other forms of expression, pointing out that the strengths of stories and games seem to be at cross purposes. Story, he says, attempts to explore internal thoughts and emotions, to blur and deepen, to get the audience to empathize with some character or situation. Games on the other hand want the player to objectify, quantize, and classify the game world. Games teach you to look at a complex system, analyze it, break it down, and put it back together again with a strategy for obtaining some goal. For many games, success requires you to look past the fiction and become an expert at some core activity. Koster suggests that games are about verbs. What I see in the market is a preponderance of games with verbs like *conquer*, *accumulate*, *maximize*, *coordinate*, etc. What would games look like based on verbs like *redeem*, *imbue*, *nourish*, *build*, *restore*. Does the market want them?

But even after we give games their due as teachers and understand what we are teaching, I think there is still something more in their God-intended purpose.

Clue #3: Games Can Be Art

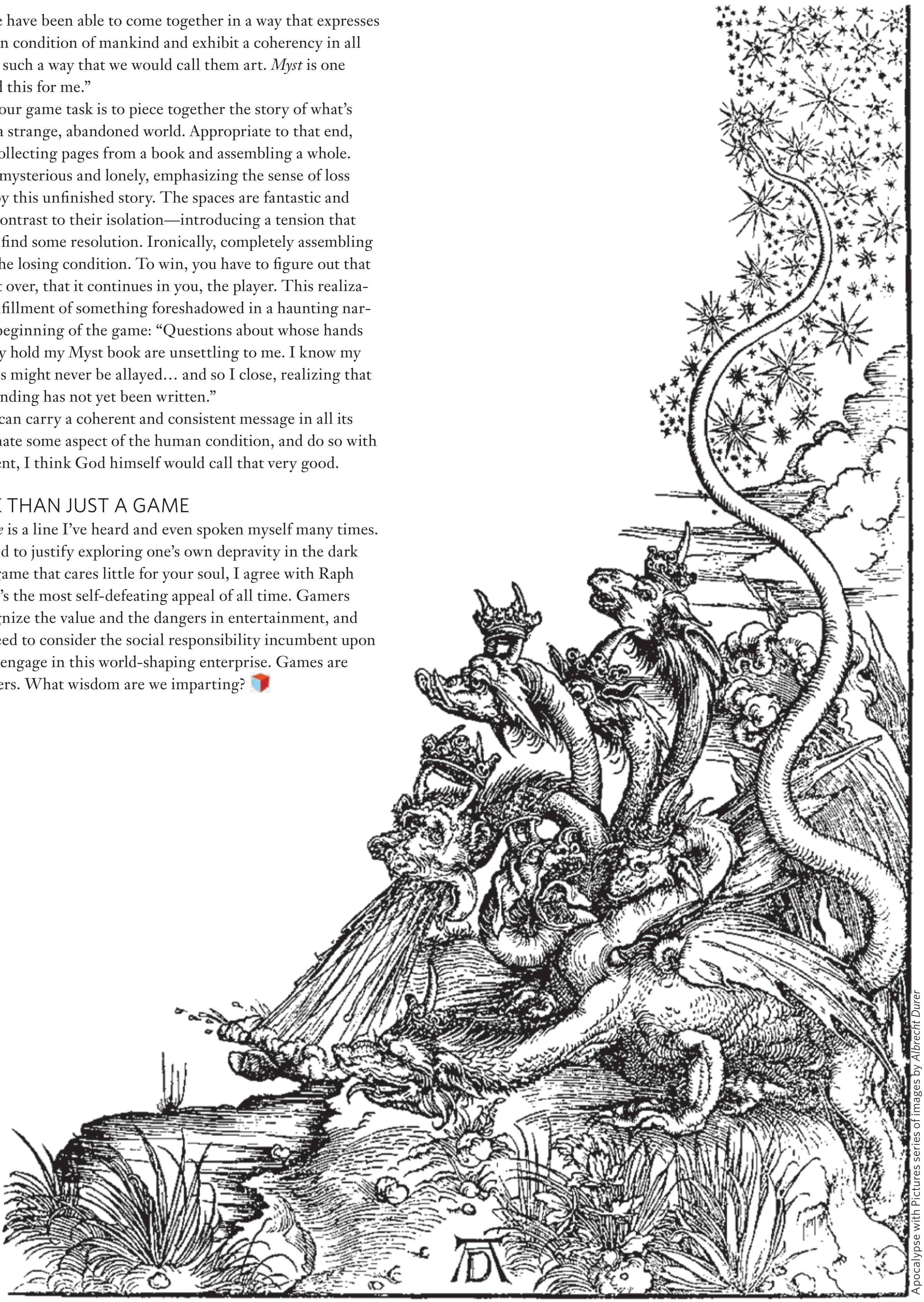
Again, in *A Theory of Fun*, Koster describes art and entertainment as terms of intensity rather than type. I'm reminded of Billy Crystal's line: "Dancing is like standing still, only faster." In any case, many of us beleaguered developers in the trenches came into the industry with an idea to create something grand. Something with some inherent lasting value. Something that touched on the human condition and spoke to someone somewhere. Something that meant something. Few games to date have been able to come together in a way that expresses some common condition of mankind and exhibit a coherency in all their parts in such a way that we would call them art. *Myst* is one game that did this for me."

In *Myst*, your game task is to piece together the story of what's happened in a strange, abandoned world. Appropriate to that end, your task is collecting pages from a book and assembling a whole. The music is mysterious and lonely, emphasizing the sense of loss engendered by this unfinished story. The spaces are fantastic and beautiful in contrast to their isolation—introducing a tension that drives you to find some resolution. Ironically, completely assembling the pages is the losing condition. To win, you have to figure out that the story isn't over, that it continues in you, the player. This realization is the fulfillment of something foreshadowed in a haunting narrative at the beginning of the game: "Questions about whose hands might one day hold my Myst book are unsettling to me. I know my apprehensions might never be allayed… and so I close, realizing that perhaps the ending has not yet been written."

If a game can carry a coherent and consistent message in all its parts, illuminate some aspect of the human condition, and do so with authorial intent, I think God himself would call that very good.

IT'S MORE THAN JUST A GAME

It's just a game is a line I've heard and even spoken myself many times. When it's used to justify exploring one's own depravity in the dark corners of a game that cares little for your soul, I agree with Raph Koster that it's the most self-defeating appeal of all time. Gamers need to recognize the value and the dangers in entertainment, and developers need to consider the social responsibility incumbent upon them as they engage in this world-shaping enterprise. Games are wisdom-makers. What wisdom are we imparting?



Apocalypse with Pictures series of images by *Albrecht Durer*

HERETIC
SHADOW OF THE SERPENT RIDERS
JOHN ROMERO'S
DAIKATANA
DOOM II
JUMPSTER!
EVIL EYE!
High Scores
SubStalker
COMMANDER KEEN
ALFREDO'S STUPENDOUS SURPRISE
GAUNTLET
SEVEN SORROWS

JOHN ROMERO

Deathkings of the Dark Citadel
Made for Hexen
Commander Keen
Aliens ate my Babysitter!
ID Software
Hovertank
Wolfenstein 3d
Anachronox
An Epic Sci-Fi RPG Adventure
Ion Storm
Eidos
Commander Keen
"Invasion of the Vorticons"
Episode One: Marooned on Mars
Doom
Floor 2 Completed

WANTS TO MAKE YOU A STAR

An Interview with John Romero

by Brenda Brathwaite

Capitol Ideas presents
Ralph Romero says - "It sucks. It's a wimpy program."
In hi-res with no sound!
Dodge 'em!
©-1982
By John Romero
Available for keyboard
Bongo's Bash II
Mach-Six
By John Romero
Capitol Ideas Soft.
Capitol Ideas presents...
Missile Defense 2
Fast arcade action by John Romero

Sketches from the early days

I've known John Romero since 1987. We met when he was working at Origin Systems on a port of *2400 AD*, and I was working at Sir-tech Software on *Wizardry V*. We grew up together, he and I, separated by genres, locations, and languages (he speaks 6502 assembler and C++, and I don't), but our collective memory is the same. We talk like kids who grew up in the same small town, knowing the same amazing characters (like Scorpia and Dr. Cat) and sharing a reverence for our founders (like Gebelli and Budge).

In a sense, though, everyone came to know John in 1993. That was the year *DOOM* was released, and his name, along with the names of the other id Software crew, became known to seemingly everyone around the world. He was featured in every form of media available to us back then, and his games, death-match antics and incredible passion for the medium were unavoidable. He helped to found and fund companies, inspired countless wannabe developers to follow their bliss, and his inherent sociability and friendliness made him the elected mouthpiece for the most successful indie game developer the world has ever seen. He was and is an impossible optimist with an incurable passion for games and a deep, deep love of programming. He has worked on 137 games, 97 of which are released. To any developer, his MobyGames entry is nothing short of humbling. It scrolls, which is more than most can say.

While his career has had its mega-ups, its very notable and much hyped downs, and its share of patently average, Romero himself has remained largely unchanged from the man I met 23 years ago. Now, like then, Romero keeps working, keeps making games, keeps trying to do something new and great. As always, John Romero wants to make the player a star.

Brenda Brathwaite: *Let's start with your design philosophy. When you make a game, what are you trying to accomplish?*

John Romero: Well, it begins and ends with the player—always has. While I am making the game, that's the role I take. I might be in coding mode or level design mode or whatever, but the only reason that I'm in that mode is because, just prior to that, I was thinking about what the player would like, what *I* would like. My process is highly iterative. I change, play, change, play, until it feels just right. Ultimately, I'm making the game for every player who likes the type of games that I like to play. So, in a sense, I am my games' audience—or the ambassador for them, anyway. That way, I never have to guess what the player might like, because it's always what I would like.

BB: *A lot of designers say that their designs are "player-centric," but can you go into it a little more detail and give us an example?*

JR: Ultimately, the player—every player—wants to be a star, and in my games, that's always what I'm trying to do: make you the star. I have succeeded and failed on occasion, I suppose, but it's always at the forefront of my mind. *Half-Life 2* did a brilliant job of making the player feel like an ultimate badass. The NPCs were in awe whenever you showed up. Valve nailed it.

BB: *Let's talk about* World of Warcraft *some. I know you're hardcore into the game, and since you have five level 80s, so does everyone reading this. You've been playing for four years now. From a design perspective, what's holding you there?*

JR: The beauty of the MMO, and especially the power of *WoW*, is that whatever mood I'm in, I can play to it in *WoW*. If I want hardcore action, I can dive into a 25-man raid in ICC. If I want to play really casually, I can level an alternate toon, level a profession, play the auction house, or just sit and talk to others. I can go to starting zones and hand out gold. I can randomly help a leveling Horde player (I'm Alliance) who's not expecting it. I can terrorize villages. The game has the largest range of activities of any game ever created. It's everything. I can always find things to do. I haven't even mentioned PvP, World PvP, Battlegrounds, or Arenas. The game is just immense.

BB: *What do you main in* WoW?

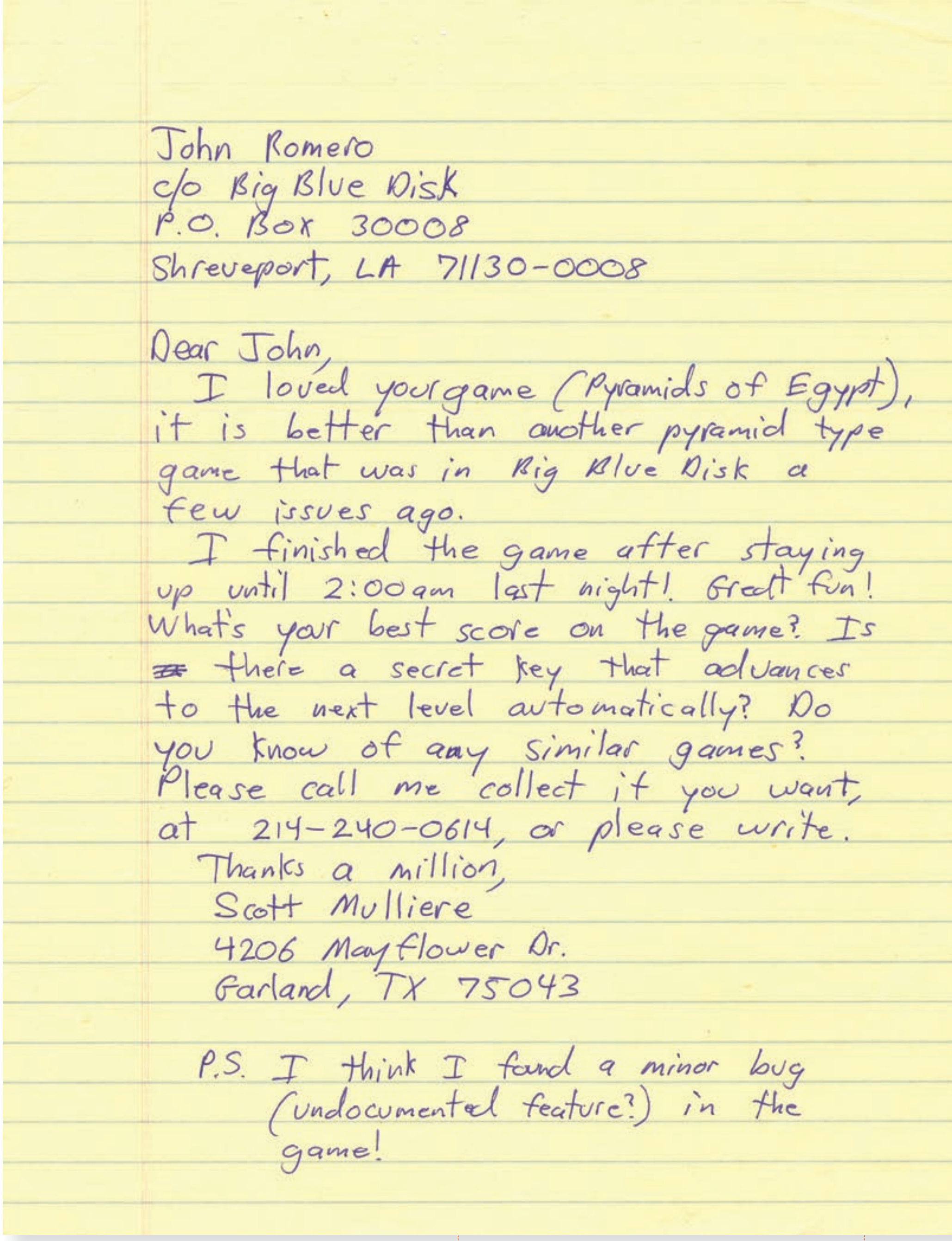
JR: Well, I have three mains, actually. Each of them are dual-spec: Holy/Shadow Priest, Protection/Holy Paladin, and Arcane/Frost Mage. I have five level 80s and those are three of them.

BB: *Your new company, Gazillion Entertainment, is also making MMOs....*

JR: I co-founded Gazillion Entertainment in September 2005 with two other guys from outside the game industry. Back then, we had a different company name but rebranded ourselves after hiring our VP of Marketing in 2008. We own four game studios, and we are developing several MMOs based on some very successful properties such as LEGO and Marvel. I became interested in the MMO soon after *World of Warcraft* was released. It was early 2005 when I saw how successful *WoW* was, and I knew for many years that MMOs have the potential to make unprecedented money. I also knew that they are the hardest category of game to create and, for me, that was a challenge I wanted to take. In addition to the pure challenge of producing an MMO, the bedrock design of my MMO was innovative and was a real design challenge.

BB: *Your current game was in the news, and it wasn't entirely clear if it was cancelled or restructured or even why, really. Well?*
JR: It's an easy answer. The market changed radically. We reached a point where it didn't make sense to continue development when we really needed to stop, analyze our approach, and target our team on a project that fits today's market rather than the market of 2008. The team continues to work diligently on the new project with a new timeline.

BB: *I think the prospect of building an MMO seems hugely daunting. Companies spend zillions of dollars to get* WoW *players to at least know their game's name, and then the players go home and play* WoW *anyway. It's hugely challenging. On the other hand, we have someone like Dr. Cat who has been working on and supporting himself off a very specialized MMO for years. Clearly, the market is broader than we give it credit for.*
JR: There's plenty of money to be made in MMOs, whether you're a *WoW* fan or not. An MMO the scale of *WoW* is not where the wider market is going. Most of the focus is on casual-persistent, and instanced designs with back-ends that don't require advanced load-balancing code at the scale of *WoW*'s servers. It won't be long before developers get really good at medium-weight MMO development and players get used to that kind of game. Console titles will have to evolve to compete in that market. Right now, when you launch a console game, you are not immediately in a world full of other players—you have to decide to connect online and find people to play with. On the PC, in an MMO, it's the only mode you play in. Going back to the core question, there are a lot of ways to make large-population online games that don't compete with *WoW*. There are very few MMOs in development or released that try to take on *WoW*. Most MMOs are not designed the same way or for the same market.


John Romero
c/o Big Blue Disk
P.O. Box 30008
Shreveport, LA 71130-0008

Dear John,
I loved your game (Pyramids of Egypt), it is better than another pyramid type game that was in Big Blue Disk a few issues ago.
I finished the game after staying up until 2:00 am last night! Great fun! What's your best score on the game? Is there a secret key that advances to the next level automatically? Do you know of any similar games? Please call me collect if you want, at 214-240-0614, or please write.
Thanks a million,
Scott Mulliere
4206 Mayflower Dr.
Garland, TX 75043

P.S. I think I found a minor bug (undocumented feature?) in the game!

Pyramid of Egypt fan letter

BB: *You've known some highs that few other developers will ever get to know. I think about the progression of something like* Wolfenstein 3D, DOOM *and then* Quake, *and that was really foundational, brilliant stuff. You even created the interface that is still standard for FPS's today. What exactly did you do on, say,* DOOM?
JR: Well, yes, for all our games at id I was usually only listed once, even though I worked on many parts of the game and company. On *DOOM*, I took Tom Hall's original designs and boiled them down to the core of what the game was about—

LETHAL LABYRINTH
MEANWHILE...
KILL THE BOY!
WE CAN'T LET HIM FIND THE EXIT!!
THERE HE IS!!
JUDY SOLVES THE RIDDLE OF THE LETHAL LABYRINTH AND ESCAPES WITH HIS LIFE... BARELY

Lethal Labyrinth sketches

pure survival. Anything that slowed the player down, I got rid of. Why did we need lives? We didn't—gone. Why score? I don't care about score—gone. Ammo, health, armor—that's it. So I did lots of game-design work, not to mention level design. I created the *DOOM* style of level design, which is very different than *Wolfenstein*. We were having trouble figuring out what the game should look like, so I took that on and figured it out. I did all of the first episode levels, except level eight. I also programmed the level editor, DoomEd, so we could make the levels. In the codebase, I programmed every flashing light, moving platform, door, stairs, damaging slime and lava, and all the other interactive world elements. I programmed loading and saving your game. I did all audio direction and programmed the music and sound effects into the game. I recorded every demo for every id game until I left. When an id game comes up and goes into demo mode, that's me playing. I posed for the *DOOM* marine character on the front of the box. I remember when our distributor had the *DOOM* boxes all made, but not ready to ship—I went to the distributor and loaded up a ton of boxes, and got them off to our first customers so they could have *DOOM* in time for Christmas. I loved what I was doing, and I had that level of involvement in every id game.

BB: *Tell us about that time. How did it feel to be on the top of the game dev world?*
JR: It was incredible. Having your game loved by the press and fans is the goal of most developers. At the time, though, we were not aware of how deeply our impact was felt because we were a very insular group and worked constantly. We had few contacts outside of our company because we pushed most people away so we could focus on our games. Getting out and meeting and playing with fans of *DOOM* was the most fun in the world. I felt on top of the world back then, but not due to the media attention. Bear in mind that I was pretty poor at a few points in my life—so this was just all so astounding to me. It was having plenty of money so I never worried about bills. I could drive awesome cars, live in a spectacular house, and have ultimate fun doing what I loved—that is the pinnacle. I still have all the qualities I had back then, with much more experience. When the right team comes around, I think magic will happen again.

BB: *You've also known some lows no other developer has known. What happened with* Daikatana*? I know you've been dragged over several sets of hot coals with this one already...*
JR: That's true. Some of those coals are deserved, though.

BB: *Like the infamous ad?* John Romero wants to.…

JR: Yes, absolutely that ad. You know, I never wanted to make you my bitch, not you, not them, not any of the other players and, most importantly, not any of my fans. Up until that ad, I felt I had a great relationship with the gamer and game development community, and that ad changed everything. That stupid ad. I regret it, and I apologize for it. I mean, there were really two products with *Daikatana*, right? One was its marketing and hype and the other was the game. You know, when the ad was first presented to me, I knew it was risky, and I didn't want to do it. It didn't make sense. I mean, there's the whole culture of smack talk that goes with games and especially FPS's, and that was something I was known for. If you death-matched me, if you even played a game of foosball against me, I was so over the top. And I wasn't alone. At id, we smashed shit—desks, monitors, keyboards, whatever. It was part of the culture at the time. So, while we all said shit to one another, it was within context. Imagine if someone from The Who went into their local music store and started smashing guitars. A lot of people would be thinking, "What the fuck? What assholes!" On stage, in context, though, it's not only okay, but expected. So when this smack-talk appeared in an ad, it was likewise out of context. Not only that, but it was something I would never say. It came off as arrogant, insulting and grandiose. I should have stopped it.

BB: *But you didn't.*

JR: No, I didn't, and I'm sorry for that. While the game could have been better on a number of levels, that ad and the hype that preceded and followed it was clearly a marketing failure and that was followed by my failure to stop it. That's what I mean by the marketing of the game as a product unto itself. I mean, just the other day on Twitter, some guy brings up *Daikatana* kneepads. I mean, seriously. There was no such thing. I have no idea where these things come from, but it's not the first time I've heard bizarre things about it.

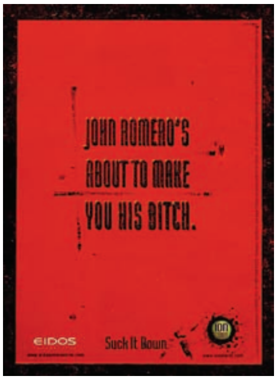
BB: *So, you say you wished you'd stopped the ad, but.…*

JR: Well, the first time I saw the ad was when our head of marketing brought it to me. His job was doing marketing for Ion, and he had this huge image campaign going on. You remember some of the ads, right? You know, film strips with each of us on it. He wanted lots and lots of hype to get us fame, attention, anything to get some kind of reaction from people. I didn't think it was necessary, but marketing and PR was his job, so I gave him room to do it. I just wanted to make games. So one day he comes up to me and says, "Here's the *Daikatana* ad we got back from the [advertising agency]." It was an interesting moment. I said, "Ah, I would never say that, though."

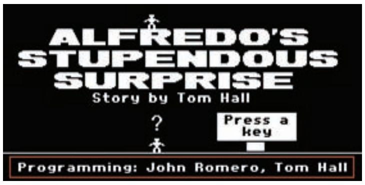
John Romero and John Carmack at id's early office.

BB: *Never say what?*

JR: First of all, "I'll make you my bitch," and secondly, just randomly out of context like that, never mind that it's to a giant audience of potentially millions of people. While death-matching, we said all kinds of garbage, but I would never tell people that I was going to make them my bitches. Ah, it was such a cocky ad. I would just never say that. I let myself get convinced, I guess. He tells me that they spent a lot of time working on it, and… all these other things. Eventually, he tells me not to "be a pussy"—that's his word, not mine—and to let him run with it. Ultimately, it was the job I hired him for, and I let him do it. But that was the moment I should have said no. I should have said fucking no. I've looked back on that three-minute meeting a hundred times. The person was fired a couple months later, but by then, the ad's damage had been done. Even if I'd come out with a brilliant game, it wouldn't have mattered. The ad insulted



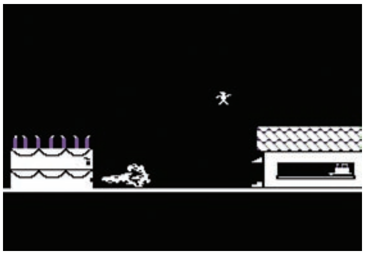
> You know, I never wanted to make you my bitch, not you, not them, not any of the other players and, most importantly, not any of my fans… That stupid ad. I regret it, and I apologize for it… You know, when the ad was first presented to me, I knew it was risky, and I didn't want to do it.


ALFREDO'S STUPENDOUS SURPRISE
Story by Tom Hall
?
Press a key
Programming: John Romero, Tom Hall

A birthday cake! How nice.

Stop right there, buster!

But I was just going to blow out the candles.



nearly everyone who read it. It was a terrible marketing decision. I apologize for letting it loose in the first place. I wish it hadn't happened, and not just for what it did to me. If I opened a magazine and saw somebody randomly smack-talking me, I don't imagine I'd relate to it well.

BB: *It affects you, though, even years later. How has it affected you personally?*
JR: Well, I'm resigned to it. There is no use in challenging and fighting everyone that puts the game down. I mean, it's only one of my games. I've made several games after *Daikatana* and a bunch before. So, it doesn't really hurt anymore when I read negative comments about the game—but sure, it did. Making *Daikatana* was like any creative work, and like it, love it or hate it, you own it and care about it. At this point, it's useless to be upset, and I am generally just not that kind of person anyway. I think the challenging time was in the beginning when the ads broke and people had this visceral reaction to them. It confirmed my suspicions, but I couldn't undo it.

BB: *It's the ad, I think, that set the stage for how the game was ultimately received. For my thesis, I actually brought up* Daikatana *with 100 different people to get their responses. I counted 100 negative responses in a row, and not one of those people had actually played the game.*
JR: That doesn't surprise me. That ad set such a tone that only a game like *Quake* or the original *DOOM* could have saved it. And even then, they still would have been upset with me. Ironically, I regularly get emails from people saying that they enjoyed the game and asking questions about it. I'm not saying it didn't have issues, and those were mainly people having trouble targeting the robotic mosquitoes and frogs on level one. [*They are harder to shoot than normal, big robots because it's an expert-level FPS.*] The other issue with the game was that the sidekick AI wasn't particularly robust. Ultimately, like many games, it had its issues. Its largest issue wasn't its game-play, though. There were so many issues with *Daikatana* that people don't know about. It's a miracle the game was released. I mean, you've made games under some challenging circumstances, I know. Everything was going wrong…

BB: *So, what happened with it? What went wrong?*

JR: In a normal development world, you work your ass off and make a fun game. You do as good a job as you can do, and give it everything you've got. All your energy goes into the game. I breathed *Wolfenstein 3D* and *DOOM*, and I loved every minute of it. So, that's a normal development world. *Daikatana* is the game you make during a war. There's no need to go into it now, I mean, this happened years ago, right? But suffice it to say that there were issues on the inside and the outside. Half the team left. With that kind of stuff surrounding the game, I wasn't working directly in the game. It was a time of tremendous personal stress, and that's not the best way to make a game. It's a miracle it was released at all. I mean, all kinds of things can go wrong at a company, but ultimately, and to coin that stupid phrase, the buck stops with me. I made some bad decisions in marketing and in hiring. Those decisions kept me from making a great game. But Ion Storm did make some amazing games, like *Deus Ex* and *Anachronox*, and I remain proud of the people and the products we produced there—including *Daikatana*.

"For my thesis, I interviewed 100 people about Daikatana. I counted 100 negative responses in a row, and NOT ONE of those people had actually played the game." – Brenda Brathwaite.

BB: *Yeah, so when I look at your ludography, you're all over the place in every genre and every platform. I know people associate you with the FPS, but that is really just a fraction of what you've done. This desire of yours to try new things, did that factor into your break with id? Did you want to do something other than an FPS in the vein in which you had been work-*

ing? It seems that id kept going in that same direction.
JR: I didn't leave id because I didn't want to make FPS's anymore. Rather I left because the company no longer had the tight collective hive-mind that it used to, and that created arguments about our creative direction. I wanted to do something really different with *Quake*. When I left and created *Daikatana*, that was more of an effort to avoid risk by sticking to the FPS formula and evolving it a little. Making another FPS and evolving it seemed less risky than doing something completely different. After Ion Storm, I made totally different games again.

"Ion Storm made some amazing games, like Deus Ex and Anachronox, and I remain proud of the people and the products we produced there—including Daikatana."
- John Romero

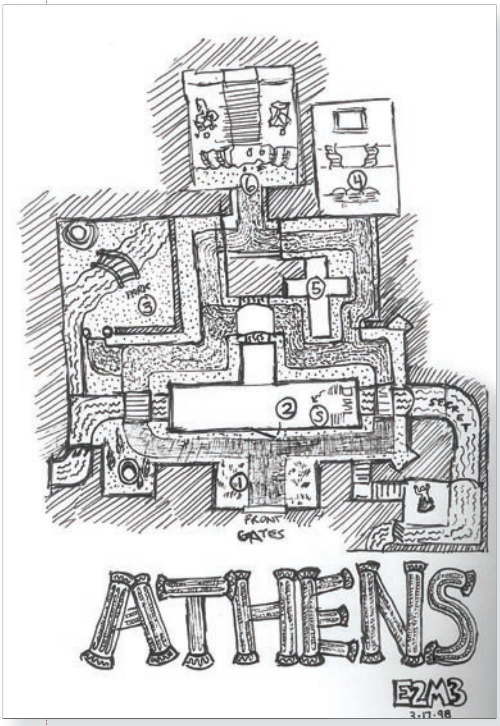
BB: *Let's talk a bit about current development opportunities. Some developers see Facebook and iPhone development as the much smaller, bastard child of the consoles. Others see it as the next big thing. What's your take?*
JR: Anyone who laments the good ol' days (back in the '80s) when a lone programmer could hammer out 6502 games and make bank needs to wake up and realize it's right here, right now. There are so many mobile devices that one coder can take advantage of. Through my 30-year career, there has always been a market for lone programmer games. Always.

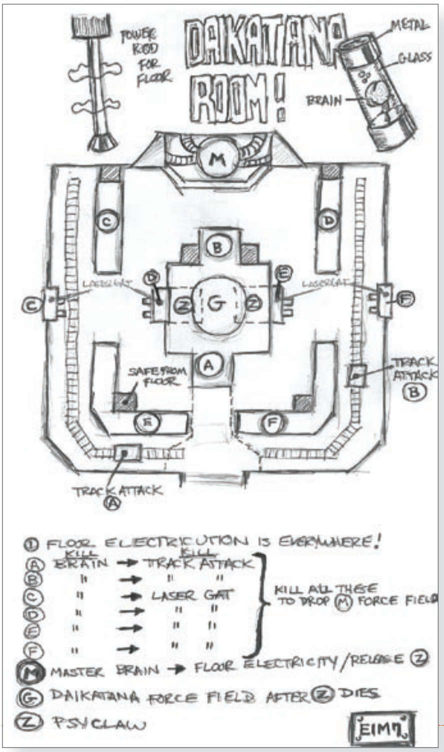
I think mobile devices are definitely the future. While at Monkeystone, I was formulating strategies about how large IP holders should be implementing designs of those properties in the mobile space. Looking down on mobile game development or social game development is patently ridiculous. Some hardcore game developers may not see the market, but that is slowly changing. The incredible money to be made in social games and mobile games is waking up certain people and companies.

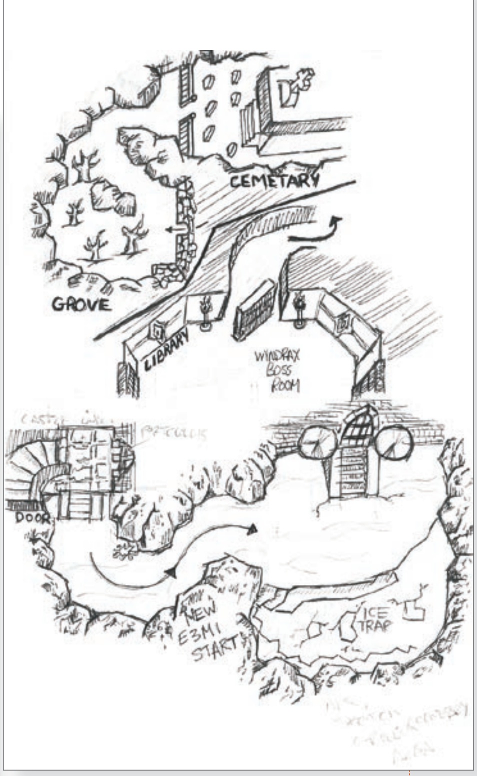
Think about this: A huge, 100-person development team makes a game for the 360/PS3/Wii that costs $40 million. Let's say you're on that team. You get a decent salary. And that's about it. Your game goes out and sells millions of copies and the company says, "Hey, thanks, you're fired," or (if you're lucky) "Hey, thanks, start on the sequel." Contrast that situation with the scenario of making your own Facebook and iPhone game (integrated together), putting it on the App Store and Facebook, and making yourself several thousand a month. Let's say that game took you three months. You can pump out four of these per year, and you get to manage everything yourself. No getting fired. No wondering about royalties. It's all yours. Wake up whenever you like. Work only with your best friends. How is that not a dream job compared to the typical situation? How can any rational person look down on that way of life? Granted, that success is not as easy as I imply here. You need a great game first, but the lure of this type of development—the rapid iteration, the small teams—is highly attractive to many.

Anyone who laments the good ol' days (back in the '80s) when a lone programmer could hammer out 6502 games and make bank needs to wake up and realize it's right here, right now. There are so many mobile devices that one coder can take advantage of. Through my 30-year career, there has always been a market for lone programmer games. Always.

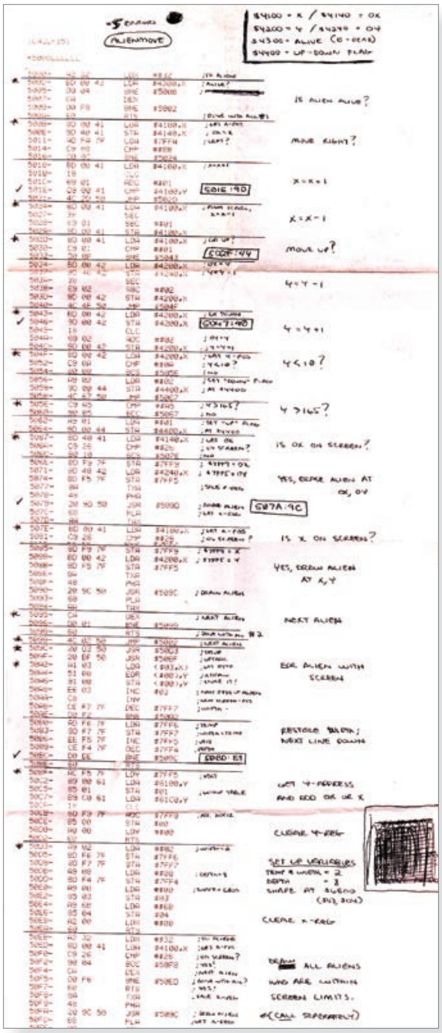
BB: *Do you think you arrived early with Monkeystone?*




POWER ROD FOR FLOOR
DAIKATANA ROOM!
METAL
GLASS
BRAIN
LASER GAT
SAFE FROM FLOOR
TRACK ATTACK B
TRACK ATTACK A
① FLOOR ELECTRICUTION IS EVERYWHERE!
A KILL BRAIN → KILL TRACK ATTACK
B " → " "
C " → LASER GAT
D " → " "
E " → " "
F " → " "
KILL ALL THESE TO DROP M FORCE FIELD
M MASTER BRAIN → FLOOR ELECTRICITY / RELEASE Z
G DAIKATANA FORCE FIELD AFTER Z DIES
Z PSYCLAW
E1M7



Daikatana level sketches


-5 errors
ALIENMOVE
$4100 = X / $4140 = OX
$4200 = Y / $4240 = OY
$4300 = ALIVE (0 = DEAD)
$4400 = UP-DOWN FLAG
IS ALIEN ALIVE?
MOVE RIGHT?
X=X+1
X=X-1
MOVE UP?
Y=Y-1
Y=Y+1
Y<10?
Y>165?
IS OX ON SCREEN?
YES, ERASE ALIEN AT OX, OY
507A:9C
IS X ON SCREEN?
YES, DRAW ALIEN AT X,Y
NEXT ALIEN
EOR ALIEN WITH SCREEN
RESTORE DEPTH; NEXT LINE DOWN
GET Y-ADDRESS AND ADD OX OR X
CLEAR Y-REG
SET UP VARIABLES
TEMP & WIDTH = 2
DEPTH = 8
SHAPE AT $6000 (90, 80)
CLEAR X-REG
DRAW ALL ALIENS WHO ARE WITHIN SCREEN LIMITS.
*(CALL SEPARATELY)

Mach-Six source code for moving aliens

JR: In a sense, we arrived too late. I remember talking to Mitch Lasky at GDC in 1999 and he told me he was going to leave Activision and start a mobile game company. He did that and started Jamdat in 2000, probably a year and a half earlier than I started Monkeystone. Jamdat produced some of the earliest mobile games (*Snake* and *Gladiator*) on archaic handsets, but in 2005 Mitch sold the company to EA for $680 million.

Monkeystone's purpose never was to become a huge mobile game developer/publisher. I just wanted to make games with Tom [Hall] and be able to code all day long. The mobile arena was really exciting to me, and I wanted to explore it. I really didn't want to deal with the business aspect personally, and that's the reason Monkeystone never really took off. I ported our games to other platforms (PC, PocketPC, Mac, Linux, GBA) to get more revenue, but we just didn't have the business experience needed to really rocket ourselves to the next level.

BB: *They say you have near perfect experiential recall of game-play experiences going back to the 1970s—but that you can't remember the name of the guy you're going to meet for dinner on Saturday. Consequently, your memory is now the subject of some big study. Were you even aware that your memory was somehow "special"?*

JR: Well, growing up and learning how to code and make games, I definitely didn't think my memory was any different than anyone else's. But as time goes on, I notice how much people have forgotten, and it doesn't make sense to me because it's so fresh in my mind, the things I've seen and done. If my memory were made of DRAM chips, it would have failed probably 10 years ago, even with constant current applied. But the human brain can remember everything if you recall it often enough—even with just a little refresh now and then. And that's what I've done over the years: telling my stories, remembering games and people's names, and caring for the history of my chosen art form.

BB: *Do you think your memory has helped you make games?*

JR: My memory has definitely helped my games, because when you're trying to innovate, you really need to know if you are innovating and not just reinventing. Knowing so much about so many games makes it easy for me to cross-check new ideas against old ones.

BB: *You've taken interest in non-digital games. Do you think the non-digital art-game you've worked on is something you will eventually release?*

JR: Yes, I think so. And there will most likely be more non-digital games—just because it's something that interests me, and I haven't done it before. Maybe I can do something different there.

> When I started ION STORM, it was as a design-first studio, and that was when I declared, "Design Is Law."

BB: *What is your single greatest moment in the industry?*
JR: My single greatest moment had to be the release of *DOOM*. But there are so many other moments close to that for me: like programming in Objective C on NeXTSTEP computers for the first time—it blew me away. Or cranking out the game *Dangerous Dave PC* in one month with all three video modes supported and changeable at any time, and seeing that 73K game survive for decades and enjoyed by so many people (I still get fan mail about it). Then there's the first time I got the *Hyperspace Delivery Boy* game map up on the screen and scrolled it around by dragging my stylus on the PocketPC screen. Or maybe even the first lo-res pixel I put on the screen on my Apple II in 6502 assembly language back in 1982.

BB: *People look at the work you did with John Carmack, and they see the brilliance there. I think they sense that if you and John could only work together again, there would be magic once more. So I have to ask: Will you and John ever make a game together again?*
JR: I have no doubt that if John and I decided to make another game it would be fresh and new and awesome. We've both grown a lot in the past 14 years and have a lot more experience, not only in game development but also in dealing with people and game teams. Many times you hear of musical groups getting back together after decades apart, and when they try to work together again everything explodes and falls apart because of personality issues and hubris. I really don't think that would happen with John and me.

BB: *"Design is Law" is a quote attributed to you. What do you mean by that?*
JR: Ever since Pac-Man, I have known that design is the real, true reason why we play games. Sure, a game must have a technology that is appropriate, and not out of balance with the design. They work hand in hand. Close to the end of *Quake*'s development, I saw the writing on the wall. All the hard programming that Carmack and Abrash did for *Quake* was going to be replaced by 3D video cards—all the rasterization of textured polygons with lighting, the calculations, etc. I knew that rendering engines were one of id's strengths, but I knew competition was going to come fast and hard after 3D was available to every developer. So what differentiates us at that point? Design. It always has. When I started Ion Storm, it was as a design-first studio, and that was when I declared, "Design Is Law."

BB: *It's been over 30 years, John. So many people burn out. Why are you still here?*
JR: I'm here because I love playing games, I love making games, and I love learning new things. If I wasn't learning something new every time I made a new game, I would not be here today. There's also a lot of reward in sticking around, and learning so much, and integrating it into everything I already know. In addition to the new, I also value very highly where we came from. I like being able to speak across the entire range of the industry with authority, because I've been there and done that. Ultimately, it's because I love games, I love making them. And, you know, I want to make the player a star.

> Ultimately, it's because I love games, I love making them. You know, and I want to make the player a star.

I shut off the recording on my iPhone, and we're done. He hands me a controller and says, "Want to play?" It's the original *Chrono Trigger* running on a tiny television, and the controller and the world feel at once both welcoming and wonderful. I notice there is a wall of code on the iMac to my left. I look at it, and he smiles. I take the controller, and he watches me play, reminding me of the parts of the game I have since forgotten. We are in his office, surrounded by games, by artifacts of his career, and by games the world has yet to see. I give the controller back to him after a while, and I watch him play. There is something wonderful here, I think: that anyone could be around this medium long enough to have embraced its success and its sorrows, its fans and its developers, and still so genuinely, deeply love games.

IN A WRITER'S PERFECT WORLD

by Rhianna Pratchett

I debated whether to use the term *writer* in the title of this article, even though it's the shorthand summary of what I do. As my industry years tick on, I've come to realize that applying the term to the act of creating narrative in games isn't completely accurate—or at least it shouldn't be if narrative professionals are being used to their full potential. Story in games is about much more than just those pesky, wordy bits, which writers are often hired to sprinkle on (when someone remembers). Done right, *story* is about building a game-world in which narrative saturates every facet—not just what is spoken or read.

For now I will use the term *writer*, but I ask that you keep the above caveat in mind as I mentally draw for you what I believe to be a writer's perfect game development world. It's important to note that this perspective is heavily writer-biased, offered in contrast to the norm merely in the hope of encouraging realistic change.

IN THE BEGINNING THERE WAS THE WORD

It starts as worlds often do, right back at the beginning, in the vastness of space where an embryo of a game idea explodes into being, and begins to take on shape and substance. Ideally that's where a writer would be. Because that's where narrative creators *need* to be: at the genesis. (Okay, so from a personal perspective, maybe I could cope with being left out of the initial big bang, just as long as I was there early enough to be able to say a few years down the line, "I remember when this was all Nazis with laser beams!")

If you get a hundred game writers into a room (and I'm not entirely sure that there *are* a hundred game writers), the one thing they will agree on is that writers are often brought into the development process too late. I hear about far too many scenarios in which the developers have all but built the story house prior to bringing in an *actual* writer, leaving only enough room for the poor wordsmith to tinker with every 23rd brick. Even worse than that is *Weekend at Bernie's* narrative creation. This is where a writer is hired so late in the project that all they can do is put a bit of make-up on the story and jiggle it around a bit, trying to give it a sense of life.

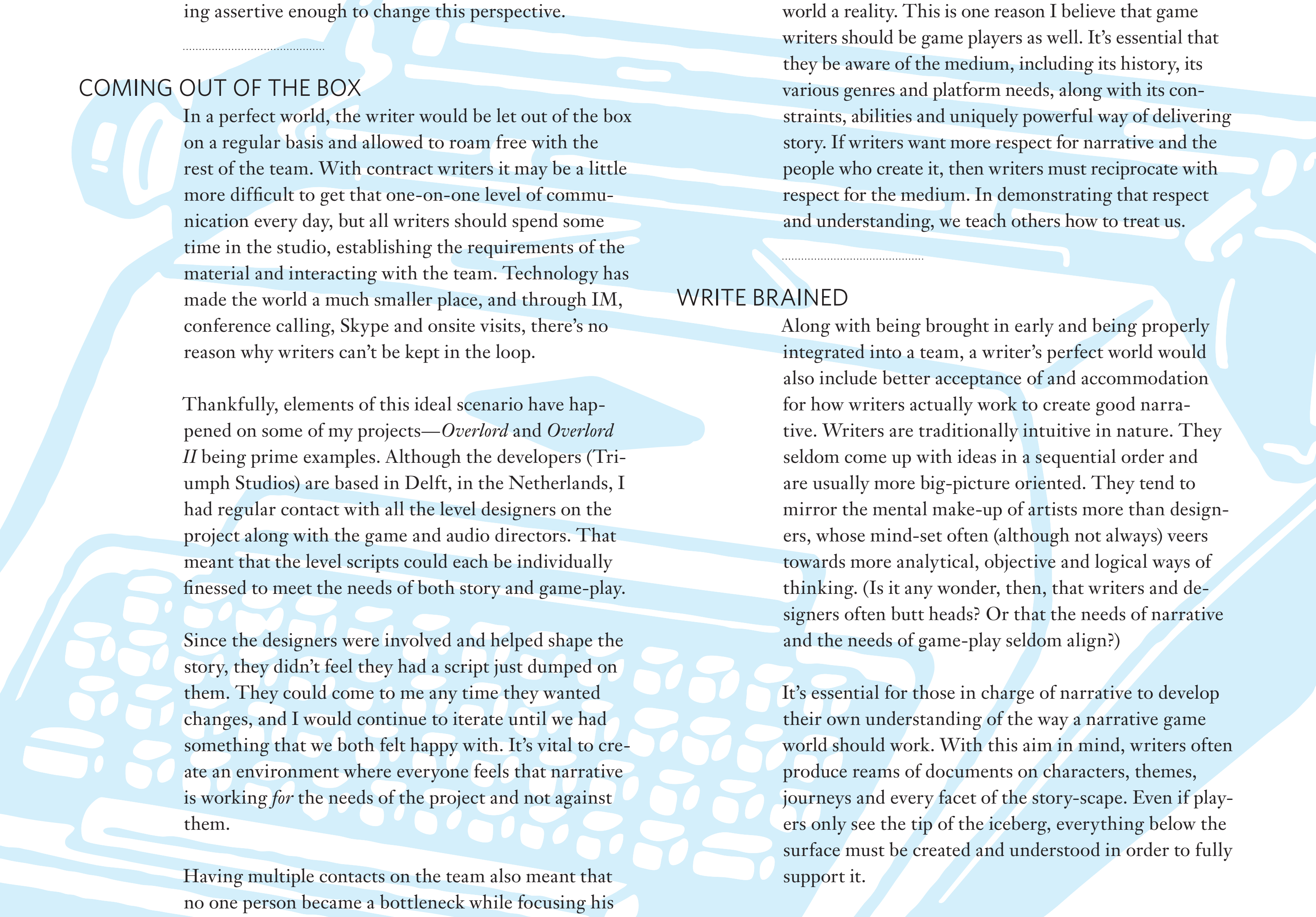
Viewing narrative as something that can just be slotted in is one of the fundamental reasons why story in games has often garnered a reputation akin to being the weird kid in the playground who collects gravel and smells of ham. A surprising number of developers I've met, talked to, heard of, and even (occasionally) worked with, have regarded narrative with a mixture of bemusement, skepticism and sometimes even barely-concealed hostility. No other facet of game development generates such a response. Perhaps that is why I get so many inquiries asking me how much it will cost for me to write X thousand lines and X other documents in an X timeframe. They might as well be starting their queries: "Hello, story robot!" I understand that there's some box-checking going on here, but treating writing as if it is something where all the Xs are pre-defined and obvious for every project and genre suggests a fundamental lack of understanding about how narrative should work. I can't help but wonder: If story isn't respected at its creation, and given adequate time and attention, then how do we ever expect it to be respected when it comes out the other side?

IT'S NOT JUST WHAT YOU'VE GOT—IT'S HOW YOU WIGGLE IT

It's somewhat comforting to see that the questions I'm often asked about game narrative have moved on from "Do games need proper stories?" to which the industry's answer seems to be a resounding "Yes"—at least for titles in which story is established as an important component. It is, however, extremely disturbing that I still get asked whether games need professional writers. If you have narrative in a game, then why would you ever consider that it should be done by someone who is not a professional writer? It's like getting the man who fixes the vending machines to do your programming.

But that aside, the role of the game writer is on the rise, which brings me to what I think is one of the most pertinent issues concerning their evolution: What to do with them once you've got them. Writers can be valuable tools (and multi-faceted ones at that) but if they aren't wielded correctly the results can be disastrous for all involved.

Writers are too often kept on a project's periphery, shut away from the team and even the game itself. It's as if developers believe that they can put instructions and money into one end of a little writer-box and words will come out the other. This mainly happens with contract writers, although I've certainly heard of it happening with staff writers as well. I can only deduce that it occurs because companies aren't entirely sure how to work with writers, or worse—because writers themselves aren't being assertive enough to change this perspective.



COMING OUT OF THE BOX

In a perfect world, the writer would be let out of the box on a regular basis and allowed to roam free with the rest of the team. With contract writers it may be a little more difficult to get that one-on-one level of communication every day, but all writers should spend some time in the studio, establishing the requirements of the material and interacting with the team. Technology has made the world a much smaller place, and through IM, conference calling, Skype and onsite visits, there's no reason why writers can't be kept in the loop.

Thankfully, elements of this ideal scenario have happened on some of my projects—*Overlord* and *Overlord II* being prime examples. Although the developers (Triumph Studios) are based in Delft, in the Netherlands, I had regular contact with all the level designers on the project along with the game and audio directors. That meant that the level scripts could each be individually finessed to meet the needs of both story and game-play.

Since the designers were involved and helped shape the story, they didn't feel they had a script just dumped on them. They could come to me any time they wanted changes, and I would continue to iterate until we had something that we both felt happy with. It's vital to create an environment where everyone feels that narrative is working *for* the needs of the project and not against them.

Having multiple contacts on the team also meant that no one person became a bottleneck while focusing his or her attentions elsewhere. While the person responsible for overall game direction should be a regular contact for a writer, making him or her the *only* contact is fraught with danger. Scripts often need constant attention, in line with the rest of the project, so having a person who can provide the writer with daily updates is incredibly beneficial.

Pairing a writer with a specific narrative designer can often have good results in getting successful integration between game-play and narrative. For those who are less familiar with the role, narrative designers are usually responsible for defining the mechanics by which the narrative is delivered, along with helping to create the story and working as a general sounding board and writer wrangler. After working with a great narrative designer on one of my current projects, I'm a complete champion for having that role in place.

But creating this perfect world cannot be accomplished unilaterally. It is not just up to the developers to define a writer's role and how they are used within the project—the writers share the responsibility for making the perfect world a reality. This is one reason I believe that game writers should be game players as well. It's essential that they be aware of the medium, including its history, its various genres and platform needs, along with its constraints, abilities and uniquely powerful way of delivering story. If writers want more respect for narrative and the people who create it, then writers must reciprocate with respect for the medium. In demonstrating that respect and understanding, we teach others how to treat us.

WRITE BRAINED

Along with being brought in early and being properly integrated into a team, a writer's perfect world would also include better acceptance of and accommodation for how writers actually work to create good narrative. Writers are traditionally intuitive in nature. They seldom come up with ideas in a sequential order and are usually more big-picture oriented. They tend to mirror the mental make-up of artists more than designers, whose mind-set often (although not always) veers towards more analytical, objective and logical ways of thinking. (Is it any wonder, then, that writers and designers often butt heads? Or that the needs of narrative and the needs of game-play seldom align?)

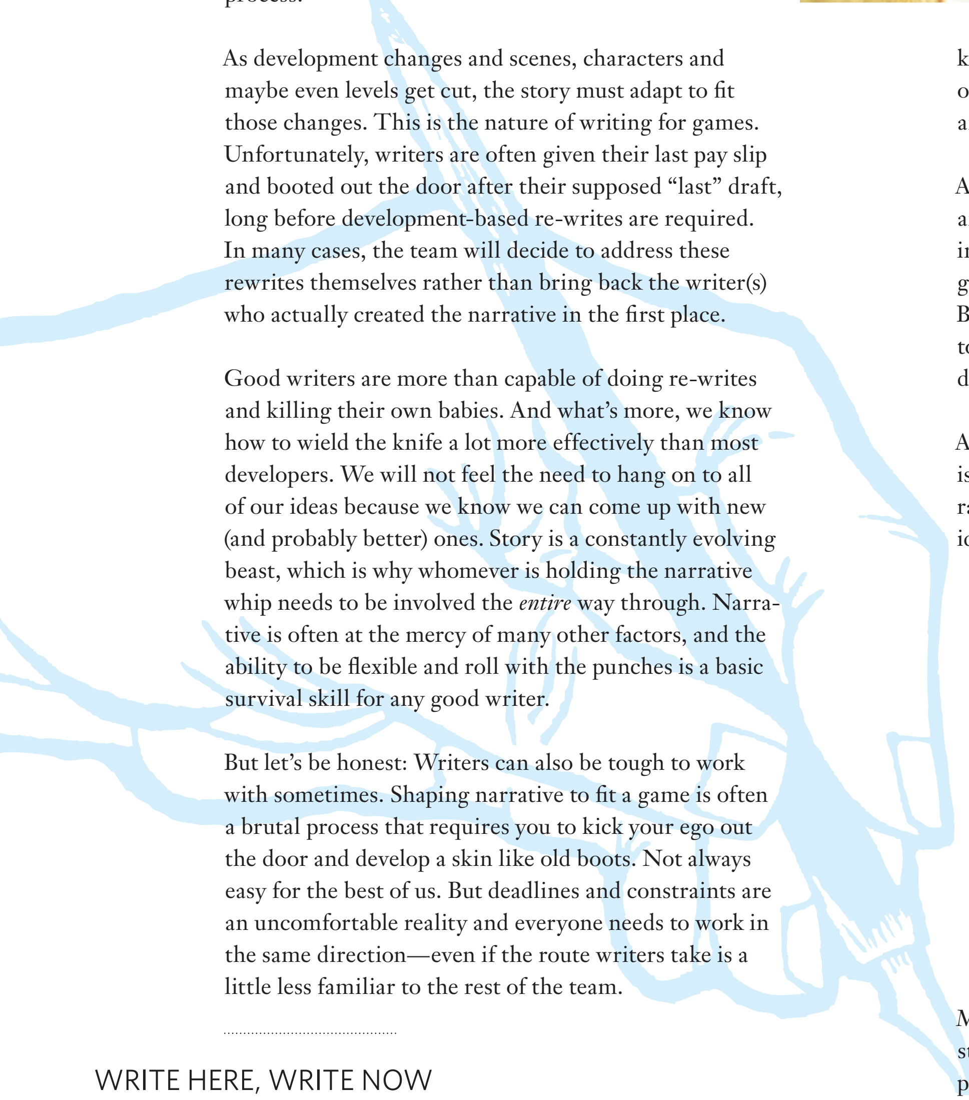
It's essential for those in charge of narrative to develop their own understanding of the way a narrative game world should work. With this aim in mind, writers often produce reams of documents on characters, themes, journeys and every facet of the story-scape. Even if players only see the tip of the iceberg, everything below the surface must be created and understood in order to fully support it.

But be warned: Often team members are not prepared for the narrative deluge, and consequently documents can largely go unread (a problem I'm constantly trying to address). The solution seems to be: reduce, reduce, reduce. Game writers usually start by writing a lot and then transcribing it into more palatable forms. If this fails to get team members to read the narrative, then perhaps you should consider laminating your documents and putting them on the backs of the doors on the restroom stalls.

CREATION AND CULL OF NARRATIVE BABIES

A vitally important component of a writer's perfect world is more time given to the actual writing process. Ideally, bringing a writer on sooner will allow that to happen, but far too many developers seem to believe that creating solid narrative ideas should be done with the speed it takes to simply flap your hands at a keyboard and type words—any words.

Writing is re-writing, and nowhere is that more the case than in games. Even the most prolific Hollywood screenwriters don't get things right first time; they draft and re-draft. But in the games industry, writers are given shockingly little time to actually write, and often they find themselves working within a timeframe drawn up by someone who has no knowledge of the writing process.



As development changes and scenes, characters and maybe even levels get cut, the story must adapt to fit those changes. This is the nature of writing for games. Unfortunately, writers are often given their last pay slip and booted out the door after their supposed "last" draft, long before development-based re-writes are required. In many cases, the team will decide to address these rewrites themselves rather than bring back the writer(s) who actually created the narrative in the first place.

Good writers are more than capable of doing re-writes and killing their own babies. And what's more, we know how to wield the knife a lot more effectively than most developers. We will not feel the need to hang on to all of our ideas because we know we can come up with new (and probably better) ones. Story is a constantly evolving beast, which is why whomever is holding the narrative whip needs to be involved the *entire* way through. Narrative is often at the mercy of many other factors, and the ability to be flexible and roll with the punches is a basic survival skill for any good writer.

But let's be honest: Writers can also be tough to work with sometimes. Shaping narrative to fit a game is often a brutal process that requires you to kick your ego out the door and develop a skin like old boots. Not always easy for the best of us. But deadlines and constraints are an uncomfortable reality and everyone needs to work in the same direction—even if the route writers take is a little less familiar to the rest of the team.

WRITE HERE, WRITE NOW

Since I first started working in the industry back in 1998, things have definitely improved for game storytelling and storytellers. I've been lucky enough to see glimpses of my perfect world on many projects I've worked on, but this is certainly not the norm. This brings me back to my opening assertion that game writing is about more than script. It's something more akin to a Swiss-army-knife, a multi-faceted craft that touches on many aspects of game creation, from script and design to animation, art and audio.

At the moment, the industry is underutilizing its writers and compromising its storytelling potential. There are individual exceptions, and this year's WGA award for games script-writing definitely celebrates some of them. But if story is an integral part of the game then it needs to be treated with the importance of any other part of development.

And that's really what I believe a writer's perfect world is all about: Respect. Respect for narrative, for the narrative process, and for narrative creators. To summarize, ideally that should manifest in several ways:

- *Time* – Getting a writer in early and actually allowing enough time to write and develop the narrative of the game world.
- *Agency* – Giving your writer a key role in a game's narrative creation (and culling) rather than merely hiring him or her to pen *your* brilliant story.
- *Attention* – Making sure that narrative is never neglected at any time throughout the development cycle and that it has a dedicated team of professionals backing it up.

My view of a writer's perfect world isn't about pushing story to the forefront or saying it should usurp gameplay; it's about weaving it together with game design and truly recognizing it as a valuable component of the creation process and overall gaming experience. This is what *has* to happen if we seek to make games like *Bioshock* and *Mass Effect* more commonplace. Such stellar games are not created by accident—those responsible have come to embrace what game writers have long been pitching: Story matters.

ouverture de session

線上

accesso

LOGIN
CONFERENCE

MAY 2010, SEATTLE, WA

注册

prijava

senha de entrada

ЛОГИН

Anmeldun

ログイン

www.loginconference.com

ntrada al sistema

온라인

CASUAL GAMES HEAD EAST

Eastern Europe Becomes Prime Location for Development and Distribution

by Tatiana Chernova

According to the Casual Games Association, game developers in Russia, Belarus and Ukraine have produced between 30 and 40 percent of the PC casual games on the international market today. As of 2010, Eastern European developers are well-represented on the lists of the top-selling games on the largest North American portals including Big Fish Games, RealGames, AOL, Yahoo! Games, MSN and others. Moreover, casual games are not by any means the only example of the success of Eastern European developers. Russian software companies also are enjoying good sales in the West, including programs for text recognition, dictionaries, machine translation, anti-virus protection, and other solutions.

HISTORY OF EASTERN EUROPEAN GAME DEVELOPMENT

It hasn't always been this way. Only 15 years ago, the list of world-famous Russian computer games contained just one title with a funny reversed letter in its name: *TETЯIS*, created by Alexey Pazhitnov. Diehard fans of the Slavic socialist gaming industry could find only one other Russian game, *Su-27 Flanker*, from Moscow-based Eagle Dynamics; but at the time, there was no Eastern European game market to speak of.

That situation would soon change.

In 1998, when Western dominance of the gaming industry seemed unshakable and the market for casual games had not yet taken shape, an economic crisis hit Russia, leading to a sharp drop in the prices of the CD versions of games. These low prices, in turn, made the development of large games economically impractical. Many developers from the post-Soviet states turned to online sales of small games which could be written quickly and could turn a quick profit (thanks to customers in the West). The pioneers in this area were the Russian companies Alawar Entertainment, KraiSoft Entertainment and Puzzle Lab. Today, over 100 developers and publishers in Russia, Belarus and Ukraine are taking this approach.

At the same time, the North American market was finally beginning to emerge. The releases weren't called casual games, however, but shareware. One of the difficulties that teams from the former USSR encountered was that, as a rule, Western publishers wouldn't work with Russian developers or take them seriously. (Almost ten years would pass before this would change.) This attitude was, in large part, due to the fact that independent developers in the former Soviet countries would create games in their spare time, meaning their work was perceived as a hobby. However, when the first attempts to sell casual games online were successful in 1999 and 2000, companies in Russia started to treat the market more seriously and turned their hobbies into full-time businesses. As a result, the first professional teams specializing in the development, publishing and distribution of casual games appeared.

ALAWAR: A RUSSIAN CASE STUDY

The history of Alawar Entertainment, for example, began with a big—but unsuccessful—retail strategy game titled *Svarog*. At first, everything was coming together well, but then the financial crisis forced the developers to release an unpolished product.

The situation was typical for the time. The economic crash buried the hopes of many programmers who were dreaming of becoming the next Richard Garriott or Sid Meier, and the Russian gaming industry all but died. However, these adverse conditions taught people how to make games without spending six figures to produce them. In order to fill their niche in the world market, without the support of large publishers, Russian developers relied on their resiliency, strength and abilities to survive and to create games in challenging conditions.

At the time, Alawar consisted of five game enthusiasts-turned-employees, who worked more for pleasure than for money. However, a move into the publishing space brought the studio its long sought-after success, and by 2002, Alawar's specialists had figured out how to optimize their cash flow and minimize risks. This breakthrough led to the publication of the three-dimensional breakout game, *Magic Ball*, which was not only a commercially successful project but also a genuine breakthrough for Russian game development. *Magic Ball* put Alawar on the map in foreign markets and showed the company's Western partners how lucrative cooperation with Eastern European studios could be.

ASSESSING THE EASTERN EUROPEAN DEVELOPMENT OPPORTUNITY

The development of a casual game is not a cheap pleasure, as making one typically costs $50,000 to $500,000. But consider this: In Eastern European countries (Russia, Belarus, Ukraine), a game can be made for half that amount without sacrificing any of its world-class quality. Case in point: Such hits as *Farm Frenzy*, *Adventures of Robinson Crusoe*, *Fishdom*, *Supercow*, *GardenScapes* and many others were developed in Eastern Europe.

There's nothing magical about this. The quality of the products Russian developers produce can be traced back to the level of education afforded Eastern European developers. As the most literate country in the world, the USSR produced a large number of high-caliber engineers, mathematicians and programmers. Today, Russia is first in the world in the number of science and engineering students, contributing 50 percent of the students in those disciplines worldwide. (By comparison, the U.S. and Japan have contributed 20 percent each).

One of the most important characteristics to note of the casual games industry at the start of the 21st century is its mobility—game development today can be done virtually anywhere. The intellectual resources in demand in the West, for example, are located near numerous scientific centers in Russia, Belarus and Ukraine, including the cities of Krasnoyarsk, Novosibirsk, Irkutsk, Nizhny Novgorod, Dnepropetrovsk, Kaliningrad and others where universities and technical colleges are located.

Likewise, it is not necessary for a developer to be located in Moscow, Kyiv or New York in order to make a commercially successful product. When developing *Farm Frenzy*, for instance, Kirill Plotnikov, Vice President of Publishing at Alawar, worked in Novosibirsk, Russia while keeping in constant contact with the developers at Alawar Melesta in Minsk, Belarus—a six hour flight away. The same approach was used in porting the Nintendo DS version of *Farm Frenzy*. Part of the team labored in Siberia, while another part worked in Bangkok, Thailand (where NorthPole Studio is located). The distance did not affect the quality of the product. The world has become smaller thanks to the Internet.

DISTRIBUTION IN EASTERN EUROPE

Meanwhile, the Western casual games market has become oversaturated, and growth rates have begun to slow. The solution? Develop international markets! To begin working online, a company needs a portfolio of successful casual games and the ability to adapt them to local markets, a process that includes managing payment methods, localization and distribution channels. The industry is ready for this kind of expansion and is already getting good results. The Russian market for casual games, for example, grew by 500 percent in 2007 and by 160 percent in 2008. In 2009, Alawar began to sell casual games in Poland, Ukraine, Finland and several other countries, and in the near future it will introduce its games to the people of the Czech Republic and Israel.

Does the Slavic game development industry have a future? I believe it does, primarily because the workforce in Eastern Europe is well-trained. Just look at the results of the International Olympiads for computers! At the ACM ICPC programming championship last spring, three out of four gold medals went to Russian teams. So have no doubt that Russia, Ukraine and other Eastern European countries hold more undiscovered Pazhitnovs capable of changing the gaming industry.

Cityscape image by Will Marantzer

PRODUCERS :

Essential Glue for any Project or Useless Bags of Meat?

by Kenn Hoekstra and Dan Magaha

The job title "producer" has no consistent meaning in video game development. Ask anyone who works in the industry and you can be sure they'll have their own unique idea of what a producer is and what he or she does. To some, "producer" is a title worthy of great respect and admiration. To others, "producer" is a four-letter word.

The reality is that production responsibilities vary at every game development studio and at every game publisher. Producer duties can range from low-level gopher and note-taker to spreadsheet-wielding taskmaster to high-powered battlefield commander.

For this article, we will focus on producers who work at the developer level in a project management capacity. So what do these producers do? According to Blizzard Entertainment Senior Producer Alex Mayberry, at a very high level, a producer's job is to "ship the product on time, within budget and of the highest quality, with 100% retention of all team members after the product ships."

Oh, is that all? Piece of cake, right?

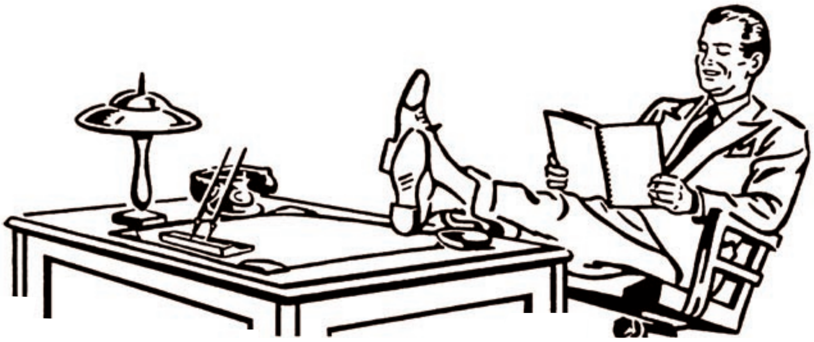
FIVE KEY PRODUCER ROLES

Developer producers play many roles, great and small, over the course of a project. Five of the most critical roles include:

- Project Manager
- Champion
- Communicator
- Counselor
- Problem-solver

There are, of course, many other duties a producer may perform, and producers often share responsibilities with other team leaders in fulfilling these roles. That said, a good producer will excel in one or more of these five areas, helping to keep a project on track and its team healthy and focused. Producers who fare poorly in these key roles get in the way of production, creating roadblocks and bottlenecks rather than removing them. Unfortunately, when producers do their jobs well, it often goes unnoticed—everything "just works." A bad producer, however, can grind even the most talented team to a screeching halt.

Let's look at these five key production roles and how, in the course of performing them, a good producer can push a team and a project to great heights and a bad one can simply push them off a cliff.



1. PROJECT MANAGER

If you're not a producer, chances are you see producers as those guys who fiddle around in MS Project all day (or, if you're one of those new age shops, those who plan "sprints" and manipulate Post-It notes on a whiteboard.) To be sure, a lot more goes into this role than just carrying a clipboard and checking off completed tasks. A producer needs to assess and mitigate risk, plan milestones, and allocate the right people to the right tasks. While all of these are extremely important jobs, being a great producer is so much more than being armed with a Gantt chart.

Good producers manage projects by empowering their leads to make decisions and to schedule their own department's tasks and people. More importantly, a good producer holds the leads accountable for these decisions and schedules. A good producer will buy back time in the product schedule when a publisher decision or unforeseen problem causes delays. As these circumstances are often beyond the team's control, a good producer will make sure the team isn't penalized for them.

Good producers know the difference between short, focused bursts of crunch and a death march, and they recognize that personal time and trust are a team's most valuable commodities. They don't squander them.

Bad producers often give in to the temptation of micromanagement. They don't empower their leads and often attempt to function as lead designer, artist, or programmer to make decisions and put their "personal stamp" on the game. A bad producer makes schedules and scheduling decisions without consulting the leads or the team and doesn't take responsibility for the consequences of those decisions.

Bad producers are frequently coaxed into mandating crunches that don't have reasonable, quantifiable goals. These kinds of crunches often turn into death marches—net negative multipliers to productivity over the long term as they poison morale, prompt turnover and absenteeism, and increase error rates in the work that *is* performed.

2. CHAMPION

Another classic producer stereotype—albeit one which is rooted in reality—is that of the glory-hog. This is a producer who's always tap dancing for upper management, taking credit for the team's work and making innumerable promises the team will later be expected to fulfill.

Negative connotations aside, the fact is all projects need a champion—someone who sings the praises of the team and the project to upper management, the press and/or investors. Someone who lives and breathes the project, and who can demo the game in his or her sleep. Someone who will talk your ear off about how great the team is, and who can bury you with a litany of the game's features.

Producers don't always fill this role—sometimes a creative director or lead designer (or even an artist or engineer!) can be a champion. More often than not, however, the biggest champion is likely to be the producer because it's the producer who is generally reporting on progress to upper management.

Good producers accept praise on behalf of the team and never fail to recognize individuals on the team, who have been especially valuable. At the same time, the champion shields individuals on the team from direct scrutiny or criticism that comes from above. A good producer champions his team members to make sure they are being treated fairly.

Bad producers hog the limelight and throw individuals under the bus when the project isn't performing up to expectations. A bad producer is the antithesis of a true champion because he is the first to take credit, but the last to take the blame.

A champion (and good producer) accepts responsibility, but also shares praise.

3. COMMUNICATOR

It has been said that good project managers spend 90% of their time communicating. For developer producers, that number may even seem a little low some weeks! Producers need to practice and foster good communication with their teams and with all levels of their organizations for their projects to succeed. Communication skills, like any other skill, can be practiced and honed, yet many producers ignore these aspects of their jobs entirely.

Producers need to solicit input from the experts on their teams, and harvest feedback about the efficacy of development practices from a representative swath of their teammates. They must close loops, disseminate critical information, and catch those ad-hoc hallway discussions that turn into critically important design meetings. Producers who sit in their offices playing with spreadsheets behind closed doors don't do their projects or their teams much good.

Good producers listen more than they talk. They do not issue ultimatums to their teams. Good producers constantly look for new tools to improve the culture of communication and collaboration on their teams, but recognize that there is no substitute for walking the halls and talking to their team members. They touch base with their leads and individual team members regularly to disseminate important information, to gauge their concerns and, perhaps most importantly, to make sure their team members are talking to each other.

On the subject of communication, Relic Entertainment's Raphael van Lierop says: "Listen for ambiguity, and stamp it out. Be clear when communicating with others and when setting goals for your team. Don't leave goals and deliverables open for interpretation—by doing so you're putting people in a position where they will probably waste effort because they don't understand what you're looking for."

Bad producers hoard information and dispense it on a need-to-know basis in an attempt to consolidate power. Ironically, many information hoarders relish claiming to have an open door policy—because their ineffectual communication skills necessitate it! However, it doesn't take long for a team to recognize these kinds of producers, and typically one or more grassroots "back channels" develop behind closed doors and via email and IM. Savvy developers recognize these toxic symptoms as signs of an unhealthy project culture which can trigger further division or even attrition.

4. COUNSELOR

Unless you've discovered some magic combination of being extremely good, lucky and/or oblivious, odds are you've experienced "team drama" during the course of a development project. For the majority of us, the reality is that game development is a stressful endeavor that can take its toll on individuals, friendships, and even marriages. This shouldn't be a surprise when you consider the #1 reason that most of us pursue game development: a creative passion to make great games.

So what happens when many passionate people enter a construct designed to put limitations and constraints on that passion? Tough decisions and politics invariably lead to hurt feelings and misunderstandings. This emotional maelstrom is where many projects live and die.

For better or worse, producers often find themselves acting as counselors for their teams. Why? Because as project managers and champions, producers are usually in a position to make decisions and effect change. For that reason, when team members have a problem, it's not surprising that they come looking for their producer to solve it.

Good producers take the time to get to know their team members. They learn what motivates and frustrates them, what their strengths and weaknesses are, and generally what kinds of individuals they are. Good producers are willing to listen to team members to let them vent and complain. They can offer feedback and solutions when asked, but most important, they simply offer to listen. Good producers help prevent burnout by making sure team members have the time and flexibility to balance their personal lives and their health against the demands of their jobs.

Bad producers simply take their cues from classic business stereotypes. They don't care about getting to know the team because in their eyes, the team is simply a group of interchangeable resources. Some bad producers mean well, but they go overboard when wearing the counselor hat. By attempting to "fix" interpersonal issues, they too often end up making more of a mess than the one they were trying to clean up. The rule of thumb here is you can fix processes, not people.

5. PROBLEM-SOLVER

Game development is not an exact science. You can be certain that even with the most experienced development teams, no two product development cycles are alike. Even the best producers, leads and developers can't anticipate everything. Let's face it—there are going to be problems.

You're going to have problems with the publisher. Problems with team members. Problems with milestones. Problems with staffing. Problems with equipment. Personal problems. Software problems. Hardware problems. Problems that range from bumps in the road to minor detours to serious roadblocks. When these problems inevitably arise, it's often the producer's job to clean up the mess. How he or she goes about doing this is what separates the good producers from the bad.

Good producers remove roadblocks and requisition critical resources for their leads to keep the team working at full capacity. A good producer can generally try to anticipate and head off many project problems before they happen.

"Great producers see puzzles, not problems," says industry veteran, producer, and engineer Brian Hook. "Every major problem a development team has is some kind of puzzle, and you can address it many different ways. There are puzzles every day, and great producers see all of this as an interesting challenge, not as a direct assault on their being."

How does a producer identify problems? "Ask good questions," says Relic Entertainment Producer Raphael van Lierop. "Good questions are ones that expose assumptions and help clarify real problems versus perceived ones. Asking good questions is the only way you'll understand the intricacies of your project and people to the degree needed to optimize efforts, solve problems, and anticipate risks."

Bad producers are reactionary. They don't anticipate problems; they run into them, bouncing from one issue to another like a pinball in a pinball machine. Bad producers find themselves "fixing" the same issues multiple times. They take pride in "being a firefighter," rather than preventing fires by watching for smoke. Bad producers often forsake finesse in favor of brute force. Rather than a puzzle solver, they become a hammer that sees every problem as a nail.

SUMMARY

More than any other position in this business, a producer's job is almost entirely about leading people—the engines of our truly chaotic, creative medium. And while those engines are exceptionally powerful, they can also be equally unpredictable and volatile. Most of the job boils down to soft skills and management, much of which can be taught, and some of which rests on a person's innate feel for leadership.

The best producers never stop trying to learn how to do it better. They never stop looking for new approaches, techniques and tools. They constantly analyze their own performance and demand personal excellence, knowing they cannot expect more from others than they expect from themselves. They take stock of the gaps in their knowledge and experience and develop plans to fill those gaps—or they surround themselves with strong, smart people who can fill gaps for them.

Producers are truly the force multipliers of game development: a bad one can weigh down even the most stalwart of teams, and a good one can help a mediocre team raise its game. But so much of a producer's potential is tied to underlying factors like the team dynamic, past history, studio management—and of course, the individual leads and managers at a company. Even the most elite producer is likely to fail if these critical elements aren't aligned.

Regardless of whether your project is a troubled mess or a finely-oiled machine, what any producer *can* do is recognize these roles and learn to embody them (and when to put them away!).

Special thanks to Jason West, Alex Mayberry, Raphael van Lierop and Brian Hook for their contributions to the content of this article.

Jake

Hi, I'm Jake Simpson, I've worked on the Sims and arcade games and a few other things, and so that's me.

Tom

Hi, I'm Tom Forsyth. So what have I done? I've written graphics cards drivers, middleware for games and now I make graphic cards again.

Steve

Hi, I'm Steve Theodore:Technical Art Director, and I've been around for about 14 years in this business. I am currently at Bungie. I'm the Technical art director there.

TALES FROM THE TRENCHES

Three Developers Share Beers and War Stories

Three guys, sitting around, drinking beer and trading war stories. (You've done it yourself a hundred times, no doubt.) There's Jake Simpson, who has worked on the *Sims* and arcade games and a few other things; Steve Theodore, who has been in the business for about 14 years and is currently the Technical Art Director at Bungie; and next to him is Tom Forsyth, who has written graphics card drivers, shipped several games (including *Startopia*), and made middleware for games.

Ostensibly all of this is true, but you know how it is with beer....

Jake:

Picture the scene. *Sims 2* has just been announced, and EA has been stupid enough to let me talk to the press. *PC Gamer* has come into our office and they're interviewing everybody. As the lead simulation engineer, I'm there to talk to the journalists about why *Sims 2* is going to be so much better than *Sims 1*. I had a bet with one of the producers before I went in there that I could use the word "lesbian" in the actual conversation, get it into the article, and no one would be offended. So anyway, at that point in time, Will Wright's tag line for *Sims 2* was "a more bitchin' people simulator." So I go into this interview and the editor Rob Smith says to me, "So what's going to make *Sims 2* better than *Sims 1*?" I said, "Well, in essence it's a more bitchin' lesbian simulator." At which point the PR girl spit her coffee across the table, the executive producer put his head in his hands, and Rob Smith said, "And there's our headline!"

Steve:

Which, I suppose, explains your successful career in PR.

Jake:

What's funnier though is the fact that the executive producer at that precise moment decided he had to share a story of development. We had added some code to *Sims 2* that caused the heads of the Sims to track other things that came into their peripheral vision—so if you walked past the Sims they would look at you and follow you as you moved. To make this work, we had to program the Sims to essentially look at a bone in the forehead of the passerby. The problem was we had some older Sims in the build at the time that did not have this bone. So what would happen if the Sim couldn't find the bone in the forehead? Well, it would just look at the root of the object—and the root of the object was (you guessed it) the groin.

Another bug was that sometimes the Sims would start looking at these things and then not look away. We had this one scenario we were demonstrating in which there was a teenager and a really old guy. And this old guy started looking at the teenager, but the bone wasn't there—so he started staring at this teenager's ass and following this boy around staring at his ass the whole time. The executive producer said we would have kept the bug in if we had been allowed to put religion into the game.

Tom:

Yeah, well sometimes you have an entirely different sort of religious experience. *Urban Chaos* was the first game I shipped. Eidos—lovely publishers at the time—had loads of money, and they arranged this big press briefing in an IMAX cinema that had the surround sound and the huge screen and everything. They were showing off a pre-build of the game that was still very raw; it would be another year or so before it actually got shipped. And so there were lots of placeholder assets we had to apologize for, but we said, "You know, it's still pretty fun." Unfortunately, some of the placeholders we had forgotten about were some of the sounds. Rather than being proper voices, they were just members of the dev team, and there were random taunts and stuff like that. Also the sound placement was off, so when a sound was behind the camera the volume didn't clamp to zero—instead it got even louder the further back it was. So anyway, we're playing along when one of the bad guys lets fly with one of these placeholder taunts—and as luck would have it, he's standing right behind the camera. So we're in the middle of this press briefing with all the big press when all of a sudden, out of these beautiful, high-fidelity, surround-sound speakers, one of the dev's starts yelling: "You wankeeeer!"

Steve:

I suppose the alternative is to wait to show your work until everything is completely ready for prime time. But if you did that, you might never get around to press briefings. For example, in 1989, when we had just shipped *Half Life*, the next big thing was supposed to be *Team Fortress 2*. Well, we didn't actually ship that one until 2007.

So we're in the middle of this press briefing with all the big press when all of a sudden, out of these beautiful, high-fidelity, surround-sound speakers, one of the dev's starts yelling: "You wankeeeer!"

Tom:

Sure, but you still managed to win best game of E3 for that game—more than once, if I'm not mistaken.

Steve:

It's true. Valve won two years in a row with *Team Fortress 2*—with a game that bore only the faintest similarity to the game that eventually shipped. We were frantic to just get something going. We had to get some momentum out there. We also didn't have any idea what the game was going to be. So we basically just grabbed two or three technologies we thought we might want to use and three or four really badly overworked people (including me), and frantically put together a *Half Life* mod that was going to be *Team Fortress 2*. The killer moment

›

for me was when we had to do this video which was supposed to be a giant battle with lots and lots of people. The battle was supposed to include like fifty people or whatever, but there was no way to script it all—so we had to do it live.

I'm there to talk to the journalists about why *Sims 2* is going to be so much better than *Sims 1*. I had a bet with one of the producers before I went in there that I could use the word "lesbian" in the actual conversation, get it into the article, and no one would be offended.

Jake:

Not good.

Steve:

I think if anyone in this room is a horrible, really evil person, you should have to be reincarnated as a guy trying to get 30 game developers to listen to you while you are piloting one client for the game, and you're trying to script all the stuff. Imagine you have 30 game developers who have been staying up all night making this crap, and they've all got grenades and rocket launchers. Eight hours of just screaming through a microphone to make a 30-second clip for the trailer. I don't think I spoke to anybody on the team for, like, two days.

That's how you get best game showing, in case you're wondering.

Jake:

Here's a not-so-award-winning moment for you. A long time ago, when I was working at Raven, we were looking at using a popular engine to build a game for Playstation 2. While in San Jose attending the Playstation developers conference, we were hanging out with all the engine guys and their famous lead guy. We went to the movies and then to dinner at Bennigan's. We were standing in the entranceway at Bennigan's; it's a noisy restaurant, and I started doing this shtick about groupies: "If our industry earns more money than the music industry and the book industry put together, where are my groupies? Right? Where's my eighteen-year-old woman throwing her underwear at me? Why is this not happening?" I'm doing this shtick and I was saying to this very famous engineer, "On the other hand, if you are into 12-year-old boys, this industry is the place to be. Right?" At that precise moment through some natural (or unnatural) force, all of the conversations in Bennigan's just sort of cease at once. And this particular individual's voice rolls out across the entire restaurant, and he says, "Yeah, wouldn't it be great if we were all pedophiles?" Epic, really.

Steve:

You want epic? I don't know if anybody here was at GDC in 2002, but there had been a lot of smack-talk between Valve and Epic about *Soul Calibur*. It ended up with a very public challenge issued by Robin Walker from Valve to Cliffy B. and his boy—they were going to have it out on the floor of GDC after hours. So we all came down there with the Dreamcast, and we conned the people from *Gamespy* into letting us use their big screen TVs. We were early, so we set up and just started playing around while we waited for Cliffy and his team. Robin, who was sort of uncannily, unnaturally, I-sold-my-soul-to-the-devil good at this game, was just taking on all comers. He played for ten minutes, then twelve, then thirty, forty, fifty minutes without losing a match. He took on, like, sixty people in a row and lost just one match. About forty-five minutes into this hour-long thing, Cliffy and his boys showed up. This was during the Cliffy B. pimp phase—with the large collar and the hat and everything. So he is there with his boys and kind of fingering his chains at the back of this huge crowd now. He and his posse start coming in towards us, and he's getting closer and closer. After he's watched about twenty-five straight wins, we're like, "Hey, you ready yet Cliff?" And we turn around and they've all gone. Just left.

Tom:

I think we have all had the experience of wanting to just get up and walk out before we get embarrassed any further. First of all, can we all agree that building jumping into games is a nightmare? [All hands go up.] Well, when we were all the way through developing *Blade 2*, we were having immense hassles with this bloody jumping thing. During run-throughs, sometimes when you jumped you could get caught in midair and just sit and spin there forever. We went through all of the hassle of implementing jumping, but 25% of the "must fix" bugs were due to this one bloody feature. It's about two days before gold master, and we've finally got the hang of all these bugs when someone says, "So, um, why do we actually use jumping then?" And everyone's like, "Well, there's levels that you have to... oh, wait, where *do* we use jumping?" And that's when it occurs to us that the only place you actually use jumping is in the tutorial where it tells you how to jump.

Steve:

You made that up.

Tom:

No, I only *wish* I had made that up.

Jake:

Does anybody here remember Rune? The Human Head game? Chris Rhinehart tells a story that happened one evening about six o'clock when he goes wandering out of his office—just mooching around, seeing what people are doing. So he walks into an artist's room, and the artist is sitting there clicking on porno sites one after the other, and Chris is standing right behind watching him do this. Chris says to the guy, "Whatcha doing?" The guy just keeps clicking through—not even looking back—and says, "Working." Chris is like "Oh." And they just stand there for a few moments, this guy clicking on dirty pics and Chris watching him. Anyway, the guy suddenly stops on this one picture of a young lady lying on the couch with her head back and her hair cascading down. The guy takes the picture, isolates it, and then grabs the image of her hair. He then posts the hair onto the clipboard, flips across to Max (the 3D editing package), and then pastes it into the model of Ragnar, the lead character on Rune. So Chris asks, "How much of Ragnar comes from porno sites?" And the guy replies, "You don't want to know."

Steve:

When it comes to boss stories, as you know I am no longer an employee of Microsoft. But our last big hurrah at Microsoft was the *Halo 3* launch. We had this huge party at our office in what used to be an old hardware store—so it's all just one gigantic room. We cleared out all the desks, and we had decorations and the giant Fuzzy Navel machine. Bill Gates is there, and so we're all crowding around shaking hands with him. About 15 minutes later, Master Chief walks in—and everybody leaves Bill Gates. So now we're all crowding around, trying to get our pictures taken with some college kid in plastic armor, while the richest man in the world is standing by himself, looking lost, holding a cup of Fuzzy Navel.

Tom:

I worked for Sega for six months in between school and university. It was kind of fun. Remember the 32X? I worked on that. It was a fun little thing but obviously didn't go well. This was at the Sega division in London, and the bosses from Japan were coming over for a visit—so we all had to be on our best behavior. We were all taught how to get the business card out and say, "Oh that's very nice." And we were told, "Whatever you do, don't put the card in your back pocket." So we all line up, and the bosses come by, and some of the guys had been practicing their Japanese. So one of them decides to use the only phrase in Japanese that he knows—something he learned from an arcade machine. He doesn't actually know what the phrase means, of course, but he did play this game a lot and so he had it down perfectly. So as the big boss comes by, he bows and utters this phrase—proper style, beautiful pronunciation, the whole bit. Turns out that what he said in Japanese means, "Now we fight—monkey style!" There is silence for like ten seconds… and then the boss cracks up. It was fantastic—except for that long, awkward ten seconds.

Jake:

That reminds me of a story a friend of mine shared with me. He used to work on *GamesMaster* magazine in the UK. He told me that many years ago they had one month in which they decided they were going get double entendres on the cover of the magazine in any way possible. They were covering a new *Street Fighter*, and one of the titles on the front cover was: "Twelve Pages of Fisting Fury." I couldn't believe they could get away with that, but he told me it wasn't even the worst one. Apparently there was an *X-Files* game coming out, and in reference to Scully their title was: "Can you penetrate her case load?" How do they get away with that stuff?

Steve:

That's one of the great things about the game industry: perpetual adolescence. It keeps us all so young and fresh looking. So about four or five years ago, they changed the file formats for all of the exporters at Bungie. They went to a particular format—called the assembly format—which was designed to allow you to group things together. So of course it became known as the ass format and ass exporter. Inevitably, from there you got to the point where when you'd instance game objects around, and that processing became known as pooping. Everybody does this at their company right? It only becomes problematic when you ship the tools to the mod makers.

So last year we shipped *Halo 2 Vista*, with the mod tools for making *Halo 2* maps. We very carefully scrubbed it for profanities and stuff, but nobody thought to look at the error dialog box in the map-making pack. So if you have a bad map file, you get a big dialog box that says in enormous letters "ass error"—along with a big picture of an engineer's butt. It was all really funny—until the ESRB got wind of it and demanded that we ship it with a "mature" label. So we had to recall a million boxes to put the "mature" label on them. We now say that guy has a million dollar butt, because that's about how much it cost us to undo that little problem.

Jake:

Ouch. That's like the porn filler game. When you build a Playstation product, quite often you physically place assets on a particular place on the disk in order to have fast access. And what happens is the application that builds the image for the disk will just fill the gaps between these files with random stuff. And as it turned out on this particular product, the application that was doing it filled this empty space with the temporary Internet cache of the guys' machine that it was burned on. And let's just say that this guy had visited quite a few websites with lots of pink on them—if you know what I mean. So apparently the disk went out there, and people started looking at the disk and found all this exciting material that was on the disk. I mean, you couldn't get to it through the game, but it was all there with lots of websites and lots of exciting stuff. A real look into the mind of a bored programmer.

We're all crowding around, trying to get our pictures taken with some college kid in plastic armor, while the richest man in the world is standing by himself, looking lost, holding a cup of Fuzzy Navel.

Tom:

That's an entirely different sort of Easter egg, isn't it?

Jake:

I love Easter eggs. I have a friend who was a level designer on a golf game and didn't like his boss very much at all. Now let me mention first that in this particular golf game, you never see the levels from any other perspective but ground level—as in you never see a bird's eye view. If you were allowed an aerial view and actually look down on all of the trees, it spells out the name of his manager and a corresponding vulgarity. Nobody ever realized it, but that's how it shipped, and it's still there inside this golf game. I'm not going to name names, but I firmly believe that you don't ever piss off your level designers because they *will* find ways to get even.

Steve:

It's also a good idea not to play against them in their own levels. We had one guy who was a really notorious secret-room stasher, and he was really good

›

at it too. Once you've done first pass on a game, you always have to go through and look for all the little self-indulgent crap that the LDs have stashed away in the corners—the stuff they're not going to tell people about. But this guy had this amazing ability to stash the hidden rooms in places where we would never find them—which came in very handy, as it turns out.

Everywhere Hitler went he would be pursued by this strange cast of characters: I think I was Sammy Davis Jr. that night, and somebody else was Joseph Goebbels in a tutu, and another guy was Bugs Bunny dressed as Stalin. Poor Hitler. We made his life hell.

I remember clearly when we shipped *Half Life*, we had a big discussion not long before we shipped about what we should and shouldn't allow people to do in the game. This was still pretty early in the days of mass Internet access, so on the advice of our lawyers, we gave users a lot of freedom to name themselves whatever they wanted, or to spray paint images on the walls, or whatever. In fact, our official position was that we exercised no control at all. We treated it like a free-speech zone.

So we had an office pool going, based on how long it would take before the first person named Hitler would show up. I won because I had said two hours, and it was actually three hours and twenty minutes after we flipped the switch and the game went live. So Hitler showed up, and we were all sitting around, having beers at the office that evening, watching to see where Hitler was going on the servers. We would just follow him, and wherever he went, we would all spawn into the same server and plaster the guy. Of course, we had all been playing the game for six months, and on top of that, the author of all the secret rooms generously revealed to us where they were. So everywhere Hitler went, he would be pursued by this strange cast of characters: I think I was Sammy Davis Jr. that night, and somebody else was Joseph Goebbels in a tutu, and another guy was Bugs Bunny dressed as Stalin. Poor Hitler. We made his life hell. We followed him around all night, and the poor guy never got over three kills because all we cared about was killing the guy and nothing else.

Tom:

Goebbels in a tutu? Are you serious? I think we've had one beer to many.

Jake:

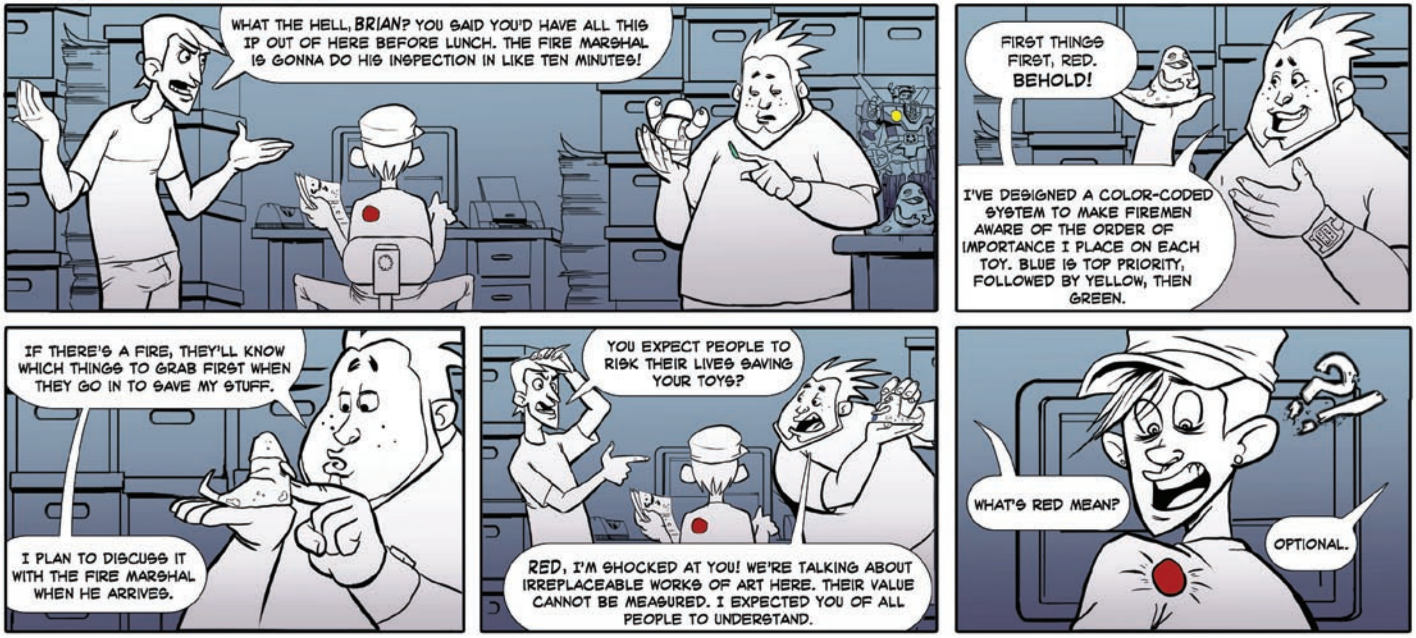
You're probably right. I'm all out of stories anyway—at least ones I'm willing to admit to on the record.

Steve:

In that case, turn off the recorder and let's order another round....

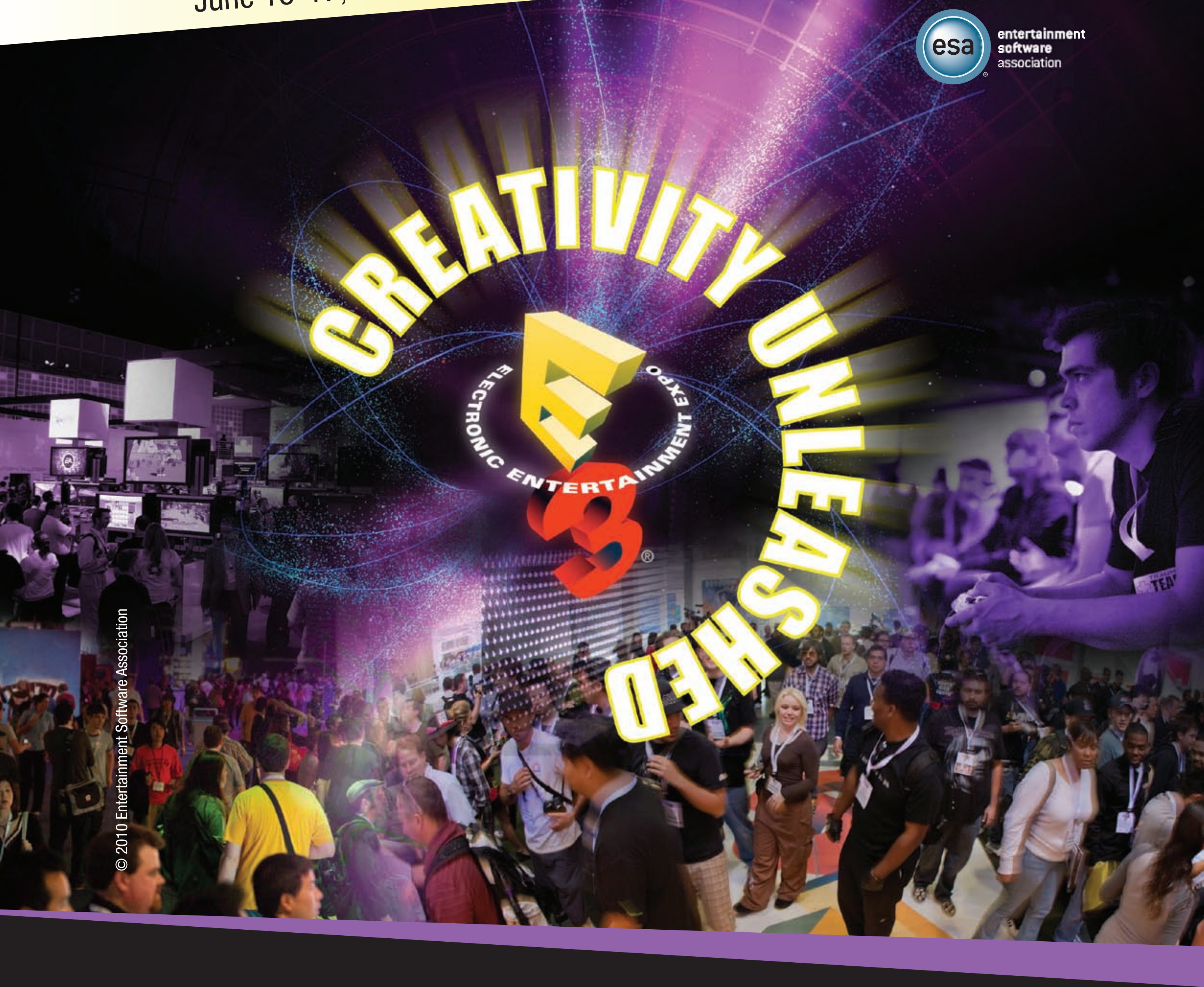
RED inc

Story and art: Ed Kuehnel and Shaun Bryant


WHAT THE HELL, BRIAN? YOU SAID YOU'D HAVE ALL THIS IP OUT OF HERE BEFORE LUNCH. THE FIRE MARSHAL IS GONNA DO HIS INSPECTION IN LIKE TEN MINUTES!
FIRST THINGS FIRST, RED. BEHOLD!
I'VE DESIGNED A COLOR-CODED SYSTEM TO MAKE FIREMEN AWARE OF THE ORDER OF IMPORTANCE I PLACE ON EACH TOY. BLUE IS TOP PRIORITY, FOLLOWED BY YELLOW, THEN GREEN.
IF THERE'S A FIRE, THEY'LL KNOW WHICH THINGS TO GRAB FIRST WHEN THEY GO IN TO SAVE MY STUFF.
I PLAN TO DISCUSS IT WITH THE FIRE MARSHAL WHEN HE ARRIVES.
YOU EXPECT PEOPLE TO RISK THEIR LIVES SAVING YOUR TOYS?
RED, I'M SHOCKED AT YOU! WE'RE TALKING ABOUT IRREPLACEABLE WORKS OF ART HERE. THEIR VALUE CANNOT BE MEASURED. I EXPECTED YOU OF ALL PEOPLE TO UNDERSTAND.
WHAT'S RED MEAN?
OPTIONAL.

June 15-17, 2010 ✶ Los Angeles Convention Center ✶ E3Expo.com

esa entertainment software association



© 2010 Entertainment Software Association

E3 Expo 2010

E3 Expo is the world's most important annual gathering for interactive entertainment.
It's where business gets done, connections are made and the future of the industry is revealed.

International Media Sponsor
MCV

Official Show Daily Sponsor
GAMEPRO MEDIA

REGISTER NOW AT E3EXPO.COM

Produced By
IDG WORLD EXPO

E3 Expo is a trade event and only qualified industry professionals may attend.
No one under 17 will be admitted, including infants. Visit www.E3Expo.com for registration guidelines.

AHA MOMENT

by Ilari Lehtinen

I have a son who is three years old and doesn't talk much yet. He's facing the challenge of picking up two languages at once, and to make matters more interesting, although the two languages sound really similar, they are completely different beasts in practice. Does he communicate, then? Oh yes. He understands bloody everything he's told, and in a way, he "talks" back with clearly understandable feedback.

I had no idea kids could reach such levels of empathy, living-along or whatever terms you might want to coin here. His body language is on such textbook-case level that it's bordering on eerie. There's clearly a communicative level of emotions and an empathic level of emotions, and they mesh together perfectly. So while his brain is figuring out the basic concepts of speech, he's developed an interim way of communication. It might be a textbook example for some people, but it gave me a hair-raising feeling of stepping on something new, opening my eyes to something quite curious. All it required was an awareness of polar opposites and some shooting in the dark. (Bear with me.)

I know people who are highly educated and use their brains on levels of analytical depth that are alien to me. On a daily basis they use vocabulary no ordinary person of daily grind has to face. The thing is, they also seem somewhat detached on a personal level of communication. Everything is questioned, referred or quoted, carrying with it an intellectual depth. To me, something feels missing when I listen to them.

It makes me wonder: Does using and knowing a great many words lead to a reduced level of empathy, to detached interpersonal communication? Going even further, will inherently "too sophisticated" and "out of their league" vocabulary simply create an emotional distance between the speaker and the audience? As your lexicon expands, do you become a sort of "textbook alien" who can no longer connect with people anymore?

At the same time, is it possible that a lack of words does not mean a lack of communication, but rather communication pushed through emotional and empathic pathways? Perhaps such communication, conveyed through body language or through the slightest twitch of some hidden muscle, is registered by the lizard brain hidden beneath our super-brain which in turn registers and analyzes things logically down to Rainman levels.

Which brings us—finally—to the subject of games.

First, think about games with a clear notion of traditional forms, lots of little details, and carefully constructed worlds in which everything connects. *S.T.A.L.K.E.R.* comes to mind. *Half-Life*. *Bioshock*. *Fallout*. These are all games with directed gameplay and new dialog, diary notes, events, and tinkering presented along the way. They communicate the game world through words of NPCs or by text. Possibly lots of it. These games achieve depth by presenting you with a barrage of information your brain begins to click through, creating coherent forms and shapes that define the world. You're encouraged to observe the world as realistic, open for literal interpretation and on-the-fly referrals—very much like reading a proper hardcover, except you have to actively stir up the book to get to the next chapters. No "automatic" flow as such. Emotions, connections and reactions occur in players' heads after interpreting the game world.

By comparison, think about games that draw you in and make you go ooh and aah over the vague feeling of intimacy of the world itself. *ICO* and *Shadow of Colossus* come to mind as prime examples. They connect with you, which means they're able to communicate with you even though they are not directly talking. They manage to deliver the very feeling of a soul throughout the game in a way you accept with open arms, dropping your defenses. They simply pull your walls down. Does this remind you of other games? Do they have lots of dialog, or do they have a distinct lack of it? *ICO* has minimal dialogue, and even then it's fictional language. Everything is delivered through actions, depicted emotions, and reactions presented to the player.

So, there we have two different ways of communicating the game world to our player and how the player should take it—two distinct approaches which (you may notice) typically do not mix within a single game. Hmmm. It's probably safer to clearly concentrate on one without mixing the other up, unless careful pacing is established to separate them in a way that supports both the game- and story-flow. Not paying attention to both approaches might tickle up that irrecoverable "something's-not-right" reaction one can't justify even if asked, meaning the game doesn't carry any impact.

It started as a string of silly musings, but as I kept wondering, it grew into this big "aha!" feeling I could reflect into my personal work ideology. It's the automatic bits of our brain that dictate how we feel about things, and I think games should have an impact of feeling just right for no apparent reason. Apparent reasons come through analytical thinking, while the feeling of "just right" comes from clicking with the game. That's empathy, regardless of genre.

The Gamesauce team, never one to work when we don't have to, decided we weren't content with ripping off Playboy and Wired, so now we are at it again—this time ripping off Oprah! She has this Aha Moment thing, which is basically where someone reveals a moment in their lives when something non obvious become obvious. Sort of like when you realize why turning your underpants inside out after you've worn them isn't a good idea. Although in this case obviously more game development like.

So here's the first one. Got a good Aha moment yourself? We want to hear about it! Although, please, no underpant stories. We got that covered.

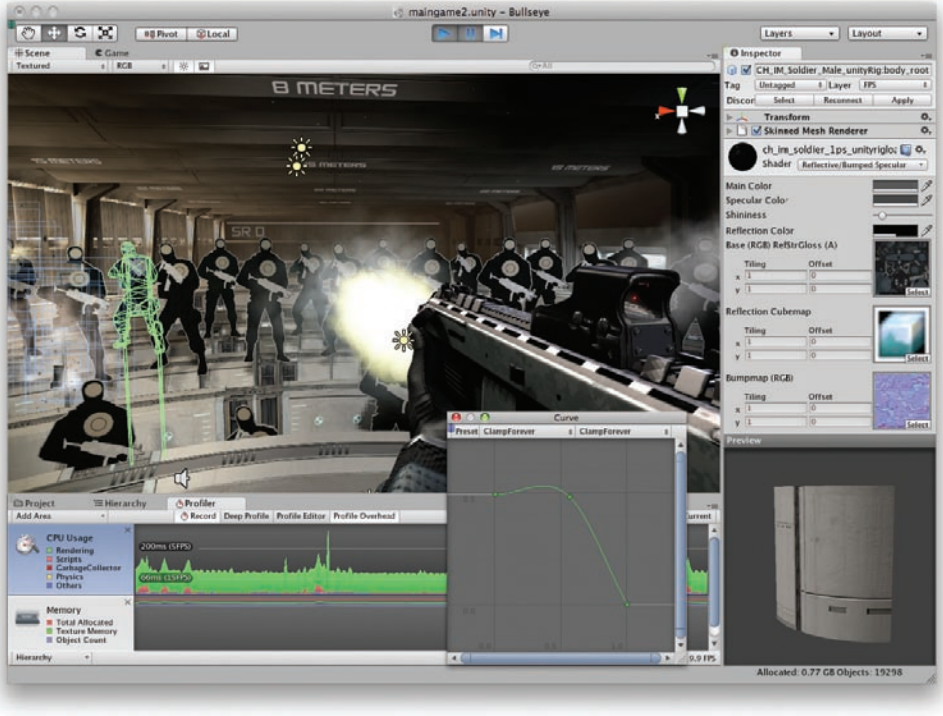
—Ed, editor@gamesauce.org

Tile art by *Invader*, photograph by *Ethan Clark*

EVEN IF YOU GOT YOUR OFFICE CHAIRS FROM THAT DUMPSTER BEHIND THE BUILDING,
your game can be Unreal.
POWERED BY
UNREAL
TECHNOLOGY

No matter what size your budget. No matter what type of game.
Unreal can be your game engine. Email Mark Rein at getunreal@epicgames.com.

Author Once, Deploy Anywhere.



UNITY3D.COM

unity

POSTMASTER:
ADDRESS SERVICE REQUESTED
GAMESAUCE
P.O. BOX 302
LAYTON, UT 84041-0302 USA